

PERSONAL COMPUTER MAGAZINE for MZ, X1, and X68000

PC

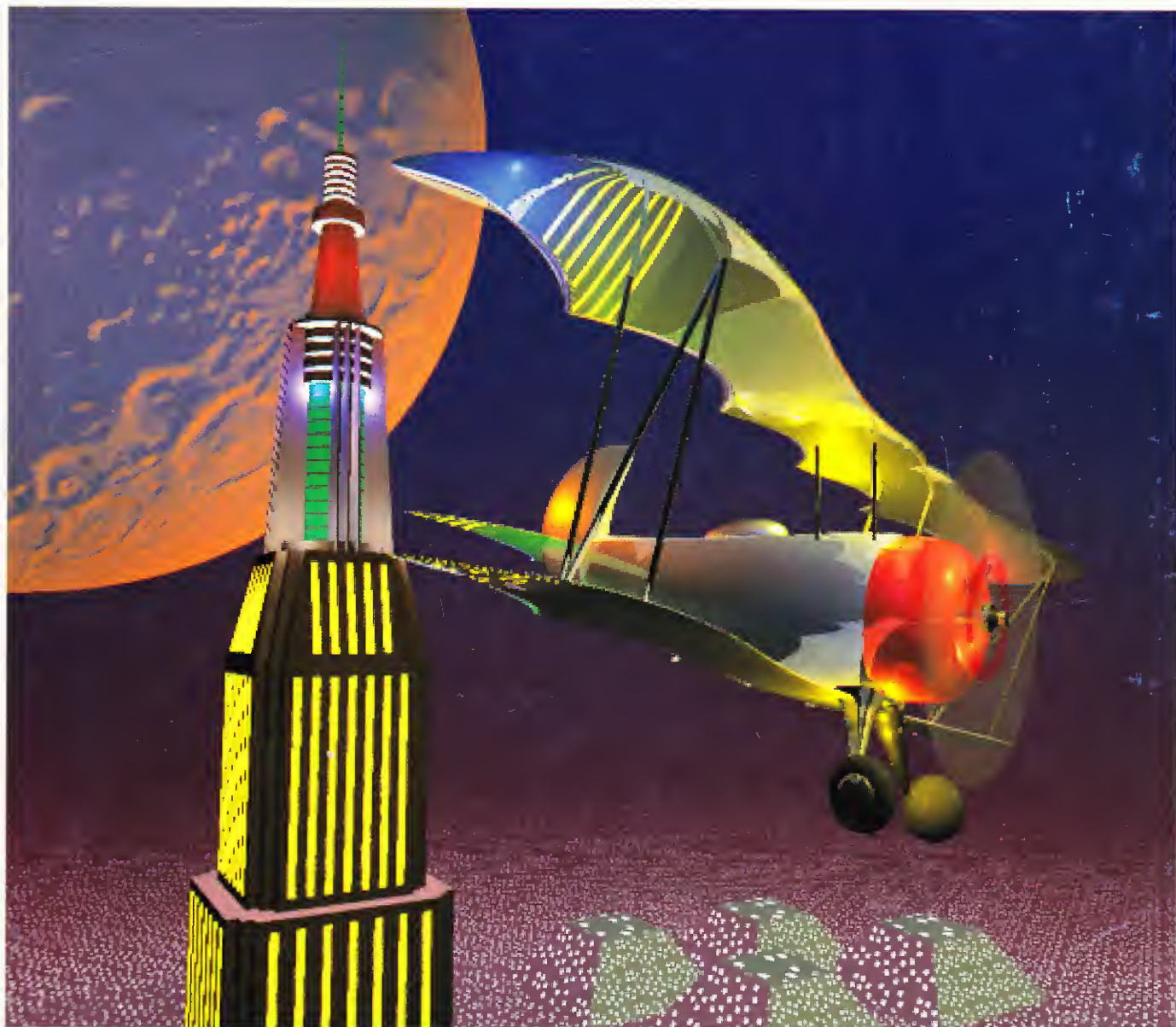
特集 ADVANCED 2D GRAPHICS

8
1990

**SOFT
BANK**

オーノエックス
定価560円

画像回転プログラムXROT0.X
X68000用カードゲームHEART
通巻100号記念特別モニタプレゼント



ひらかれた知性。



サ・ワークステーション。80Mバイトハードディスク、SCSI インターフェイスを標準装備。

SUPER HD

本体+キーボード+マウス+トラックボール

CZ-623C-TN(チタンブラック) 標準価格498,000円(税別)

アートの系譜。

EXPERT II

本体+キーボード+マウス+トラックボール

CZ-603C-BK(ブラック)・-GY(グレー) 標準価格338,000円(税別)/HDタイプ CZ-613C-BK(ブラック) 標準価格448,000円(税別)

ニュースタンド。

PRO II

本体+キーボード+マウス

CZ-653C-BK(ブラック)・-GY(グレー) 標準価格285,000円(税別)

HDタイプ CZ-663C-BK(ブラック)・-GY(グレー) 標準価格395,000円(税別)



次代のユーザーインターフェイスを象徴する"SX-WINDOW"^{*}搭載。

今回のX68000ニューシリーズのデビューに関して、ハードウェア以上にウィンドウ環境の提供に耳目が集中したことは、昨今のビジュアルユーザーインターフェイス事情をふまれば、当然のことと言えるでしょう。マルチウィンドウを駆使してX68000をコントロールする、待ち望まれていた環境がこのSX-WINDOWによって実現されるのです。何の予備知識もなしにこのウィンドウに接した方は、一見して従来のビジュアルシェルのバージョンアップと思われるかもしれませんが、本質的には全く異質のものと言えます。ひとつのウ



ィンドウである仕事をさせながら、別のウィンドウで違う仕事にとりかかる。ひとことで言えばアプリケーションを実行させる環境としてのウィンドウであるということ。これまでのビジュアルシェルではできなかったシーンを生み出しています。複数のアプリケーションを同じ操作のもとで走らせたり、アプリケーション相互でデータのやりとりが可能になるわけです。そして、次代のインテリジェンスを鮮やかに象徴する4階調のハイセンスな画面処理——。SX-WINDOWをターゲットとしたアプリケーション開発もすでに推進されており、これからの展望という点からも大いに期待されるどころです。また、このSX-WINDOWはディスクによって供給され、BIOSの高速化(平均2倍)も含めてOSであるHuman68kの機能を拡張。ニューシリーズのみならず、すべてのX68000でこの新しい環境が享受できます。

* SX-WINDOWの起動には、メインメモリ2MBが必要です。CZ-6000/6010/6110/6520/6530/6620/6630でSX-WINDOWをご使用の際には、あらかじめ別売の1MB増設RAMボードを増設してください。

NEW X68000 PERSONAL WORKSTATION SUPER·EXPERT·PRO

充実のディスプレイラインアップ

15型カラーディスプレイテレビ(ドットピッチ0.39mm)	CZ-602D-BK(ブラック)・-GY(グレー)	標準価格 99,800円(チルトスタンド同梱・税別)
15型カラーディスプレイテレビ(ドットピッチ0.39mm)	CZ-605D-BK(ブラック)・-GY(グレー)	標準価格115,000円(スピーカー2個/チルトスタンド同梱・税別)
15型カラーディスプレイテレビ(ドットピッチ0.31mm)	CZ-613D-TN(チタンブラック)・BK(ブラック)・-GY(グレー)	標準価格135,000円(スピーカー2個/チルトスタンド同梱・税別)
14型カラーディスプレイ(ドットピッチ0.31mm)	CZ-603D-BK(ブラック)・-GY(グレー)	標準価格 84,800円(チルトスタンド同梱・税別)
14型カラーディスプレイ(ドットピッチ0.31mm)	CZ-604D-BK(ブラック)・-GY(グレー)	標準価格 94,800円(スピーカー2個/チルトスタンド同梱・税別)
21型カラーディスプレイ(ドットピッチ0.52mm)	CU-21HD-BK(ブラック)	標準価格148,000円(スピーカー2個同梱・税別)

* 印の商品は在庫僅少です。

EXEリーダーズグッズ プレゼント実施中

- いま、EXE会員よりご紹介のお客様がEXEショップでX68000シリーズを購入されますと、EXE会員にEXEリーダーズグッズをプレゼントします。詳しくはEXEショップにお問い合わせください。
- また、X68000シリーズをご購入のお客様は、ぜひEXEクラブにご入会ください。

● お問い合わせは…シャープ(株)電子機器事業本部システム機器営業部 〒545大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)
電子機器事業本部液晶映像システム事業部第2商品企画部 〒162東京都新宿区市谷八幡町8番地 ☎(03)260-1161(大代表) **シャープ株式会社**



特集 ADVANCED 2D GRAPHICS



カードゲームHEART



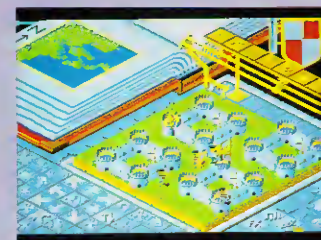
たべくずし



大航海時代



ウルティマV



プロミストランド

Oh!X

C O N T

●特集

40 ADVANCED 2D GRAPHICS

- | | | |
|----|--|------|
| 44 | X68000用グラフィックツール紹介
あなたにあったグラフィックツール | 荻窪 圭 |
| 50 | ギザギザのないグラフィック関数
アンチエイリアシングとは? | 丹 明彦 |
| 68 | X-BASICによる画像処理
後処理によるジャギーの除去 | 中野修一 |
| 72 | 色数の補間と量子化
グラフィックデータを変換する | 鈴木康弘 |
| 77 | 4096色→8色変換
Zの画像をX1で | 亀田雅彦 |

●Oh!X通巻100号記念特別企画

- | | | |
|-----|-------------------|------|
| 23 | 表紙ぎゃらりい | |
| 97 | 対戦ポピュラス 祝一平VS西川善司 | 浦川博之 |
| 100 | 愛読者特大モニタープレゼント | |

●シリーズ全機種共通システム

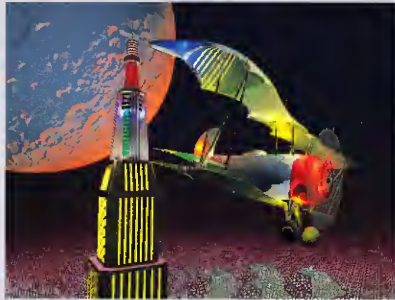
- | | | |
|-----|--------------|------|
| 145 | THE SENTINEL | |
| 146 | リンカWLK | 石上達也 |

●読みもの

- | | | |
|-----|-------------------------------------|------|
| 158 | 第40回 知能機械概論——お茶目な計算機たち——
人工知能の冒険 | 有田隆也 |
| 160 | 猫とコンピュータ 第50回
サーチャーでござる | 高沢恭子 |
| 162 | X-OVER NIGHT 第3話
旅行あれこれ | 高原秀己 |

〈スタッフ〉

●編集長/前田 徹 ●編集/植木章夫 岡崎栄子 浅井研二 ●協力/有田隆也 中森 章 後藤貴行
林 一樹 荻窪 圭 岡本達一郎 毛内俊行 吉田賢司 影山裕昭 相馬英智 古村 聡 村田敏幸 丹
明彦 三沢和彦 長沢淳博 宮島 靖 金子俊一 浦川博之 山田純二 ●カメラ/杉山和美 ●イラスト
/永沢しげる 山田晴久 小栗由香 ●アートディレクター/島村勝頼 ●レイアウト/元木昌子
AD GREEN ●校正/グループこじら



表紙絵：須藤 牧人

1990 AUG. 8

E N T S

●THE SOFTOUCH

- | | | |
|----|--|------|
| 28 | SOFTWARE INFORMATION
話題のソフトウェア | |
| 32 | GAME REVIEW
大航海時代 | 浦川博之 |
| 34 | ウルティマV | 荻窪 圭 |
| 36 | プロミストランド | 山田純二 |
| 38 | AFTER REVIEW
天下統一/ダウタウン熱血物語
あ〜くしゅ/Yet Another Column | |

連載/紹介/講座/プログラム

- | | | |
|-----|---|----------------------|
| 81 | X68000用画像回転プログラム
XROTO.X | 渡辺伸也 |
| 88 | X68000 CARD.FNC用カードゲーム
HEART・負けるが勝ち | 池谷昌彦 |
| 92 | X1turbo用ディスク管理プログラムINTEGRAL X1
トランジェントコマンドを作る | 亀田雅彦 |
| 102 | PC-E500テーブルトークRPGサポートシステム(1)
ポケコンでCARPGを | 松井 信 |
| 104 | ハードウェア工作入門(2)
基本インタフェイス回路 その2 | 三沢和彦 |
| 107 | X-BASICプログラミング調理実習(13)
超入門・ファイル処理 | 泉 大介 |
| 113 | X68000マシン語プログラミングChapter_OFH
マウスwithグラフィック | 村田敏幸 |
| 121 | PASCALプログラミングへの招待(3)
PASCALのデータ型を見る | 藤井義巳・藤木健士 |
| 126 | マシン語カクテル in Z80's Bar 第14回
楽な逆ポーランド? | 山田純二 |
| 130 | (で)のショートプロバてい その12
祝! 1周年記念 | 古村 聡 |
| 134 | Oh!X LIVE in '90
OMENS OF LOVE (X68000)
ENDLESS RAIN (X1/turbo)
ダートフォックスよりRunning up!(X68000MUSICDRVサンプル曲) | 小玉和博
伏喜義宏
西川善司 |

ペンギン情報コーナー.....164
FILES Oh!X.....166
Oh!X質問箱.....168
STUDIO X.....170
編集室から/DRIVE ON/ごめんなさいのコーナー/SHIFT BREAK/microOdyssey.....174

UNIXはAT&T BELL LABORATORIESのOS名です。
Mechはカーネギーメロン大学のOS名です。
CP/M, P-CPM, CP/Mplus, CP/M-86, CP/M-68K, CP/M-8000, DR-DOSはDIGITAL RESEARCH
OS/2はIBM
MS-DOS, MS-OS/2, XENIX, MACROS80, MS CはMICRO
SOFT
MSX-DOSはアスキー
OS-9, OS-9/68000, OS-9000, MW CはMICROWARE
UCSD p-systemはカリフォルニア大学理事會
WordStar, WordMasterはWORDSTAR International
TURBO PASCAL, TURBO C, SIDEKICKはBOLAND INTER
NATIONAL
LSI CはLSI JAPAN
HuBASICはハードソンソフト
の商標です。その他、プログラム名、CPUは一般に各
メーカーの登録商標です。本文中では"TM", "R"マー
クは明記していません。
本誌に掲載されたプログラムの著作権はプログラム
作成者に保留されています。著作権上、PDSと明記さ
れたもの以外、個人で使用するほかの無断複製は禁
じられています。

■広告目次

アイツ186
アイビット電子190
アクセス192
アンス・コンサルタンツ9
エスピーエス181
AVCフタバ電機183
オーエーランド187
OKハウス182
計測技研184・185
コナミ12・13
ザイン・ソフト11
J & P表3
システムサコム14・15
シャープ表2・表4・1・4-8
ソフトクリエイト189
九十九電機22
T & Eソフト17
デンキヤ188
パソコンプラザオクト20・21
P & A18・19
ビクター音楽産業16
満開製作所191(下)
ロゴスシステム10



ディスプレイ関連

カラーディスプレイテレビ



15型カラーディスプレイテレビ
CZ-602D-BK
★CZ-602D-GY
標準価格 99,800円(税別)
(チルトスタンド同梱)

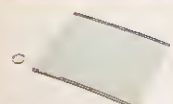


15型カラーディスプレイテレビ
CZ-605D-BK-GY
標準価格 115,000円(税別)
(スピーカー2個・チルトスタンド同梱)



15型カラーディスプレイテレビ
CZ-613D-TN-BK-GY
標準価格 135,000円(税別)
(スピーカー2個・チルトスタンド同梱)

CRTフィルター



高性能CRTフィルター
BF-68PRO
標準価格 19,800円(税別)
(14 15型用)

カラーディスプレイ



14型カラーディスプレイ
CZ-603D-BK-GY
標準価格 84,800円(税別)
(チルトスタンド同梱)



14型カラーディスプレイ
CZ-604D-BK-GY
標準価格 94,800円(税別)
(スピーカー2個・チルトスタンド同梱)



21型カラーディスプレイ
CU-21HD
標準価格 148,000円(税別)
(スピーカー2個同梱)

チューナー



RGBシステムチューナー
CZ-6TU-BK-GY
標準価格 33,100円(税別)
(リモコン付)

アートツール

画像入力



カラーイメージスキャナ*1
CZ-8NS1
標準価格 188,000円(税別)



スキャナ用パラレルボード
CZ-6BN1
標準価格 29,800円(税別)

映像入力



カラーイメージユニット*2
CZ-6VT1-BK
CZ-6VT1
標準価格 69,800円(税別)

プリンタ

カラープリンタ



24ドット
熱転写カラー漢字プリンタ
★CZ-8PC3
標準価格 65,800円(税別)
(信号ケーブル同梱)

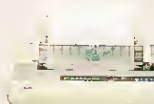


48ドット
熱転写カラー漢字プリンタ
CZ-8PC4
CZ-8PC4-GY
標準価格 99,800円(税別)
(信号ケーブル同梱)



カラービデオプリンタ
CZ-6PV1
標準価格 198,000円(税別)
(信号ケーブル同梱)

カラーイメージジェット



カラーイメージジェット*3
IO-735X
標準価格 248,000円(税別)
(信号ケーブル別売)

ドットプリンタ



24ピン
カラー漢字プリンタ(80桁)
CZ-8PG1
標準価格 130,000円(税別)
(信号ケーブル同梱)



24ピン
カラー漢字プリンタ(136桁)
CZ-8PG2
標準価格 160,000円(税別)
(信号ケーブル同梱)



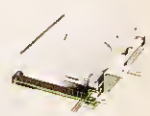
24ピン漢字プリンタ(136桁)
CZ-8PK10
標準価格 97,800円(税別)
(信号ケーブル同梱)

ファイル

ハードディスク



ハードディスクユニット(20MB)
CZ-620H
標準価格 178,000円(税別)



増設用ハードディスク
ドライブ(40MB)
(CZ-602C/603C/652C/653C内蔵用)
CZ-64H
標準価格 120,000円(税別)
(取付費別)

*取付に関してはシャープ
お客様ご相談窓口にてご
相談ください。

W・V turbo シリーズ用 周辺機器

標準価格は税別です。

カラーディスプレイ

- 21型カラーディスプレイ*1 CU-21HD 148,000円

映像・画像入力編集装置

- カラーイメージスキャナ CZ-8NS1 188,000円
- カラーイメージボードII CZ-8BV2 39,800円

- 立体映像セット ★CZ-8BR1 29,800円
- パーソナルテロップ*2 CZ-8DT2 44,800円

FM音源

- ステレオタイプFM音源ボード CZ-8BS1 23,800円

スピーカー(2本1組)標準装備、ミュージックツール同梱

プリンタ

- 24ピンカラー漢字プリンタ(80桁) CZ-8PG1 130,000円
- 24ピンカラー漢字プリンタ(136桁) CZ-8PG2 160,000円

ファイル

- ミニフロッピーディスクユニット(2HD+2D)*3 ★CZ-520F 118,000円

X68000をサポート。



シャープペリフェラルファミリー X68000



ボード

拡張メモリ



1MB増設RAMボード
(CZ-600C専用)
CZ-6BE1
標準価格 35,000円(税別)



1MB増設RAMボード
(CZ-601C/611C/652C/
653C/652C/663C用)
CZ-6BE1B
標準価格 28,000円(税別)



2MB増設RAMボード*4
CZ-6BE2
標準価格 79,800円(税別)



4MB増設RAMボード*4
CZ-6BE4
標準価格 138,000円(税別)

インターフェイス



ユニバーサルI/Oボード
CZ-6BU1
標準価格 39,800円(税別)



GP-IBボード
CZ-6BG1
標準価格 59,800円(税別)



増設用RS-232Cボード
(2チャンネル)
CZ-6BF1
標準価格 49,800円(税別)

数値演算プロセッサ



数値演算プロセッサボード
CZ-6BP1
標準価格 79,800円(税別)

FAX



FAXボード
CZ-6BC1
標準価格 79,800円(税別)

MIDI



MIDIボード
CZ-6BM1
標準価格 26,800円(税別)

ネットワーク

モデム



モデムユニット*5
CZ-8TM2
標準価格 49,800円(税別)
(RS-232Cケーブル同梱)

RS-232Cケーブル



RS-232Cケーブル
(平行接続型)
CZ-8LM1
標準価格 7,200円(税別)



RS-232Cケーブル
(クロス接続型)
CZ-8LM2
標準価格 7,200円(税別)

LANボード



LANボード
CZ-6BL1
標準価格 268,000円(税別)
CZ-6BL2
標準価格 298,000円(税別)
*電源ユニット/ソフトウェア
(ネットワークドライバVer1.0)同梱

入力



インテリジェントコントローラ
CZ-8NJ2
標準価格 23,800円(税別)



マウス・トラックボール
CZ-8NM3
標準価格 9,800円(税別)



トラックボール
CZ-8NT1
標準価格 13,800円(税別)



マウス
CZ-8NM2A
標準価格 6,800円(税別)



ジョイカード
CZ-8NJ1
標準価格 1,700円(税別)

その他



拡張I/Oボックス(4スロット)
(CZ-600C/601C/602C/603C/
611C/612C/613C/623C用)
CZ-6EB1-BK
CZ-6EB1
標準価格 88,000円(税別)

スピーカー



アンプ内蔵
スピーカーシステム(2本1組)
AN-S100
標準価格 36,600円(税別)

システムラック



システムラック
(CZ-600C/601C/602C/603C/
611C/612C/613C/623C用)
CZ-6SD1
標準価格 44,800円(税別)

*1. 本使用に際しては、あらかじめ別売の1MB増設RAMボードCZ-6BE1 標準価格35,000円(税別・CZ-600C用)、CZ-6BE1B 標準価格28,000円(税別・CZ-601C、CZ-611C、652C、653C、662C、663C用)を増設してください。
*2. モデムユニットCZ-8TM2に同梱のソフトはX1/X1ターボシリーズ用です。

●ミニプロセッサユニット(2D)	★CZ-502F	99,800円
●ミニプロセッサユニット(2D・1Dタイプ)	CZ-503F	49,800円
●増設ミニプロセッサユニット(2D)*4	CZ-53F-BK	19,800円

拡張ボード・その他

●モデムユニット(300/1200ボー)	CZ-8TM2	49,800円
●100%外部メモリ	CZ-8BE2	29,800円
●RS-232C・マウスポート*5	CZ-8BM2	19,800円
●ビデオディスプレイインターフェイス*6	CZ-8BF1	14,800円

●JIS第1水準漢字ROM*7	CZ-8BK2	19,800円
●RS-232C用ケーブル(平行接続型)	CZ-8LM1	7,200円
●RS-232C用ケーブル(クロス接続型)	CZ-8LM2	7,200円
●拡張I/Oボックス	CZ-8EB3	33,800円
●RFコンバータ*8	AN-58C	2,980円
●インテリジェントコントローラ	CZ-8NJ2	23,800円
●マウス・トラックボール	CZ-8NM3	9,800円
●マウス	CZ-8NM2A	6,800円
●トラックボール	CZ-8NT1	13,800円

●ジョイカード	CZ-8NJ1	1,700円
●チルトスタンド	CZ-6ST1-E-AB	5,800円
●高性能CRTフィルタ*9	BF-68PRO	19,800円
●スキャナ用パラレルボード*10	CZ-8BN1	27,800円

●品番中の-表示は、B<ブラック>・E<オフィスグレー>を示します。*1 X1ターボシリーズ用 *2 CZ-862Cには接続できません *3 X1ターボシリーズ用 *4 CZ-830C用 *5 X1シリーズ用 *6 CZ-850CでCZ-502Fを使用する場合に必要 *7 CZ-800C、801C、802C、803C、811C、820C用 *8 CZ-820C、822C、830C用 *9 14/15型用 *10 CZ-8NS1用 ●接続等の説明につきましては、周辺機器総合カタログをご覧ください。

★印の商品は在庫僅少です。

"アート"と呼べる高水準のソフトウェアが

(次代のインテリジェンス、 ウィンドウ環境をあなたのX68000で。)

ユーザー本位の操作環境を提供するフル画面マルチウィンドウタイプの美しいデスクトップ(テキスト面/単色4階調+カラー4色、グラフィック面/カラー65,536色中16色)、イベント・ドリブン型マルチタスク処理により複数の作業を同時に処理できる疑似マルチタスクや入出力装置の設定が簡単に行える多機能コントロールパネルを搭載した本格ウィンドウシステムです。従来のビジュアルシェルとは異なり、今後のアプリケーションソフトが統一された操作環境で実行できるようになります。



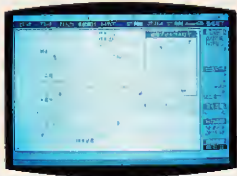
SX-WINDOW ver1.0

CZ-259SS 10万台達成ご愛用感謝価格6,800円(税別)



(高速通信をサポート。これからの、 そしてさまざまな通信環境に対応する 高機能コミュニケーションソフト。)

Communication PRO-68Kのバージョンアップ版です。300BPSから19,200BPSまでの通信速度に対応し、パソコン同士の接続や各種データベースの漢字端末に、またホストコンピュータとのデータ通信に利用できます。さらにMNPモデムへの対応で、ハードフロー制御(CTS/RTS)をサポート。その他、高速逆スクロール機能、オートログイン/オートパイロットが可能な自動実行機能、コンカレント機能も装備。行入力機能やスクリーンエディタなど豊富な編集機能も魅力です。また、バイナリファイルを転送するプロトコルとしてXmodem(128/SUM, 128/CRC, 1K)、Ymodem(G, BATCH, G-BATCH)、Transit2(TEXT, BINARY)プロトコルもサポートしています。



CZ-257CS

標準価格
19,800円(税別)

Communication PRO-68K ver2.0

(ソースコードデバッグをはじめ、 各種開発ツールを強化。 バージョンアップされたCコンパイラ。)

Cのソースレベルでデバッグできるソースコードデバッグを搭載したほか、各種開発ツールを強化した総合開発ツールです。また、ライブラリはHuman68k ver2.0の拡張DOSコールもサポートしているなど、よりX68000のハードウェアを活かせる豊富なライブラリ(約800種)となっています。強力なMAKEも新たに追加。C言語の標準であるANSI規格準拠をさらに強化し、プロトタイプ宣言もデフォルトに変更されました。「BASIC-Cコンバータ」、「アセンブラ」、「リンク」、「デバッグ」、「ソースコードデバッグ」、「アーカイバ」、「ライブラリアン」、「コンバータ」などのツールが装備されています。



CZ-245LS

7月発売予定

C compiler PRO-68K ver2.0

X68000をサポート。



シャープオリジナルソフトウェア
68000

ビジネスツール

Hyperword

■CZ-251BS 標準価格39,800円(税別)

X68000の優れたグラフィック環境を活用し効率的に文書を作成するためのインテリジェントワープロです。アイデアプロセッサ機能、ハイパーテキスト機能などをサポート。データの整理やプレゼンテーションツールなど幅広い用途に利用できます。



TOP給与計算エキスパート

■CZ-228BS 標準価格200,000円(税別)

給与計算から明細発行までを、リアルイメージ入力により自動的に、素早く処理することができます。

TOP財務会計

■CZ-227BS 標準価格200,000円(税別)

会計エキスパートシステムとデータベースを搭載し、機能と操作性を両立させた財務会計ソフト。

CYBERNOTE PRO-60K

■CZ-243BS 標準価格19,800円(税別)

プライベートなデータやビジネスデータを簡単な操作で管理・運営できるパーソナルデータベースです。リフィル、タックシール、ハガキなどの印字もOK。シャープ電子手帳とのデータ交換可能(別売の通信ケーブルCE-200Lが必要)。



CARD PRO-60K

■CZ-226BS 標準価格29,800円(税別)

自由なレイアウト画面で入力できるワープロ機能を装備したカード型リレーショナルデータベース。

CARD PRO-60K用システム手帳リフィル集

■CZ-241BS 標準価格9,800円(税別)

CARD PRO-60K用活用フォーム集

■CZ-242BS 標準価格9,800円(税別)

Stationery PRO-60K

■CZ-240BS 標準価格14,800円(税別)

他のソフトを起動する前に、このStationery PRO-60Kを一度起動するだけで、他のソフトを実行中にも「スケジュール」「住所録」など多彩な機能をワンタッチで使用できます。シャープ電子手帳とのデータ送受信も実現。(別売の通信ケーブルCE-200Lが必要)。



DATA PRO-60K

■CZ-220BS 標準価格58,000円(税別)

入力の手間を軽減するヒストリー機能を装備した、コマンド型リレーショナルデータベースです。

BUSINESS PRO-60K

■CZ-212BS 標準価格68,000円(税別)

スプレッドシート(表計算)、データベース、グラフ作成機能を一体化させた統合ビジネスツールです。

グラフィックライブラリ VOL.1

■CZ-235GS 標準価格8,800円(税別)

暑中見舞用を中心としたNEW Print Shop PRO-60K用グラフィックデータ集。

グラフィックライブラリ VOL.2

■CZ-236GS 標準価格8,800円(税別)

年賀状を中心としたNEW Print Shop PRO-60K用グラフィックデータ集。



NEW PrintShop PRO-60K

■CZ-221HS 標準価格19,800円(税別)

オリジナリティあふれるはがき等、簡単に作成、印刷できるホームプロダクティビリティツール。ほとんどの処理をアイコンで表示しマウスで選ぶフレンドリーオペレーション。

アートツール

Musicstudio PRO-60K ver.1.1

■CZ-252MS 標準価格28,800円(税別)

24トラック対応MIDIマルチレコーディングソフトMusicstudio PRO-60Kがバージョンアップしました。従来の機能に加え、小節間のコピー及びデリートや、MIDIインプットモニターなど、数々の機能を追加・改良。さらに使いやすくなりました。
※MIDIボード(CZ-6BM1)が必要です。



MUSIC PRO-60K (MIDI)

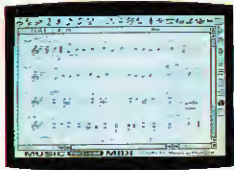
■CZ-247MS 標準価格28,800円(税別)

MIDI対応自動伴奏機能をサポート。簡単な楽譜入力でMIDI演奏が楽しめます。
※MIDIボード(CZ-6BM1)が必要です。

ソングライブラリ<101曲集>

■CZ-248MS 標準価格8,800円(税別)

鑑賞用と音楽データ加工工作用からなるライブラリです。



Sampling PRO-60K

■CZ-215MS 標準価格17,800円(税別)

AD PCM機能を活かす高性能サンプリングエディタ。多彩なEDITORを装備。サンプリング音のデータはBASICでも活用できます。

SOUND PRO-60K

■CZ-214MS 標準価格15,800円(税別)

スタジオのコンソールパネルを操作する感覚でFM音源による音創りが楽しめるサウンドエディタ。

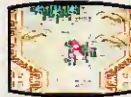
MUSIC PRO-60K

■CZ-213MS 標準価格18,800円(税別)

最大8パートのスコア(総譜)が書け、内蔵のFM音源で演奏できる楽譜ワープロ・演奏用ツール。



シューティングゲーム
「ツインビー」
■CZ-217AS
標準価格7,800円(税別)
© KONAMI 1988



シューティングゲーム
「沙羅曼蛇」
■CZ-218AS
標準価格8,800円(税別)
© KONAMI 1989



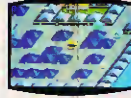
ブロックゲーム
「アルカノイド」
■CZ-222AS
標準価格7,800円(税別)
© TAITO CORP. 1987



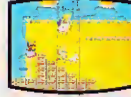
ドライブゲーム
「フルスロットル」
■CZ-231AS
標準価格8,800円(税別)
© TAITO CORP. 1988



スポーツゲーム
「熱血高校
ドッジボール部」
■CZ-232AS
標準価格7,800円(税別)
© TECHNOS JAPAN CORP. 1988



アクションゲーム
「バックマニア」
■CZ-233AS
標準価格7,800円(税別)
© NAMCO



アクションゲーム
「ニュージーランド
ストーリー」
■CZ-230AS
標準価格8,800円(税別)
© TAITO CORP. 1989



スポーツゲーム
「V-BALL」
■CZ-246AS
標準価格7,900円(税別)
© TECHNOS JAPAN CORP. 1989



バイクレーシングゲーム
「スーパーハンガオン」
■CZ-238AS
標準価格8,800円(税別)
© SEGA 1987



ジェットウィルイミュレーションゲーム
「サンダーブレード」
■CZ-239AS
標準価格9,500円(税別)
© SEGA 1987



アクションゲーム
「ダウタウン熱血物語」
■CZ-254AS
標準価格8,800円(税別)
© TECHNOS JAPAN CORP. 1989

開発ツール

OS-9/X68000

■CZ-219SS 標準価格29,800円(税別)

OS-9のもつマルチタスク機能、リアルタイム機能を活かした使い易く機能的なOS環境を提供。これまでのデータ資産も活かれます。
※OS-9はマイクロウェア社の登録商標です。

Human68k ver.2.0

■CZ-244SS 標準価格9,800円(税別)

THE 極楽V2.0

■CZ-224LS 標準価格9,980円(税別)

■AI-68K (Staff LISP/OPS PRO-60K)

■CZ-234LS 標準価格188,000円(税別)

サウンドツール

必聴、必見。

NEWミュージックトレンド“MIDI”体験!!



パソコンミュージック **MIDI**
X68000

音遊サウンドライブ

in Summer

- X68000が創造するパソコントレンド、MIDI。

音楽さえ好きであれば、楽器やパソコンが苦手な人でも即エンターティナーになれる、いま注目度一番のニュートレンドです。

- 音遊サウンドライブは、プロのキーボード奏者による本格的なMIDIライブコンサート。

好評の第2弾ではプレイングショーだけでなく、ミュージシャンの楽しいトークや、サウンドスケープ、曲あてクイズなど、X68000とMIDIの楽しさを実感して頂けるイベントがグンと増えました。

- イベント参加者には、オリジナルTシャツやX68000オリジナルグッズをプレゼント。

あなたの感性をとがらせる新鮮で活気あふれるMIDIライブが、あなたをきっと興奮の“音遊”空間へ誘ってくれることでしょう。



EXEクラブが待っている。

● X68000を手にしたら何はともあれ「EXEクラブ」へ。本体同梱の入会申し込みハガキを送るだけで会員証として、オリジナルデザインのカード電卓がもらえちゃう(会費無料)。EXEクラブニュースや最新ソフト、周辺機器などX68000の最新情報を随時ご案内。各種イベント、フェアへのご招待もあります。

(「X68000」は持っているけど、まだ入会していない方も、ぜひこの機会にお申し込み下さい。)

● EXE会員にはEXEリーダーズグッズ・プレゼントも実施中です。詳しくはお近くのEXEショップまで。



NEW X68000、新作ソフト、面白イベント……

まるごと見・体・験フェア。

● 今回のテーマはニューX68000。SUPER-HD/EXPERT II/PRO IIの魅力までにご体験ください。業界注目のSX-WINDOWも必体験。他、新作ソフト体験コーナー、100インチ液晶プロジェクションによる大迫力のゲームたちなど、新しい出会いがあるかもしれません。X68000オリジナルグッズも展示即売。ぜひお近くの会場へお立ち寄りください。

● X68000見体験フェア・音遊サウンドライブ開催日程

開催月日	開催地区	開催場所	お問い合わせTEL
7/20(金)・21(土)	東京	ソフトクリエイイトX68000フェア	03-486-6541 ◎
7/22(日)	太田	パソコンランド21太田店X68000フェア	0276-45-0721 ◎
7/22(日)	金沢	サンミュージックOAプラザX68000フェア	0762-48-6131 ◎
7/22(日)	高松	シャープ見体験フェアイン高松	0878-23-4868 *
7/23(月)	高崎	パソコンランド21高崎東口店X68000フェア	0273-26-5221 ◎
7/28(土)・29(日)	札幌	九十九電機札幌店X68000フェア	011-241-2299 ◎
7/28(土)・29(日)	富山	シャープ見体験フェアイン富山	0762-49-1181 ◎
7/28(土)・29(日)	神戸	星電社三宮本店X68000フェア	078-391-8171 ◎
8/4(土)	高崎	パソコンランド21高崎駅前店X68000フェア	0273-64-0521 ◎
8/4(土)	京都	J&P京都寺町店X68000フェア	075-341-3571 ◎
8/5(日)	前橋	パソコンランド21前橋店X68000フェア	0272-21-2721 ◎
8/5(日)	姫路	星電社姫路本店X68000フェア	0792-88-1717 ◎
8/5(日)	高知	シャープ見体験フェアイン高知	0888-83-5522 *
8/11(土)・12(日)	宇都宮	計測技研新設開店フェア	0286-22-9811 ◎
8/12(日)	伊勢崎	パソコンランド21伊勢崎店X68000フェア	0270-21-3121 ◎
8/12(日)	東京	T-ZONE X68000フェア	03-257-2650 ◎

◎印の会場で音遊サウンドライブを開催します。＊印の会場には山下真実出演。

!!**シャープ株式会社**

●お問い合わせは…シャープ株式会社電子機器事業本部システム機器営業部
 〒545大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)

'90 7月27日(金)

新規OPEN!!

CG画像制作センター 秋葉原サテライトオフィス

●新住所

〒101 東京都千代田区外神田6-3-8 外神田田島ビル3F

TEL 03-839-8481 (但し、7月20日より通話可能) (JR秋葉原駅徒歩5分)

—— アンス・コンサルタンツ東京本部事務所(現高輪)は7月15日をもって上記へ移転します。 ——

●オープニング見学会

'90 7月27日(金) 11:00~14:00 お取り引き先、マスコミ、他一般
15:00~20:00 ユーザー様

主な業務案内/CG画像制作プロデュース・アプリケーション開発受託・サイクロンユーザー会ネット
ワークサポート・3次元CAD×CGシステム導入コンサルタント及び教育・アウトプットサービス等々

制作スタッフ募集!! CG画像制作センター

CGプロダクション(仮称:アトリエ68)
として、CG制作ユーザー会・関東支部を開設します。
ユーザーの方はどしどし制作スタッフ登録をして下さい。

※申し込み方法その他詳しくは福岡本社までお問い合わせ下さい

'90 第2回サイクロンCG大会 9月24日に決定!!

全サイクロンシリーズユーザー対象(98、68、TOWNS)

静止画、アニメその他サイクロンを使用した作品なら何でもOK!!

●作品受付期間 8月10日~9月8日(当日消印有効)

●賞金・クランプリ 20万円、その他賞金・賞品多数

※詳細は近日発表!!



サイクロンExpressα(エクスプレスアルファ)好評発売中!!

サイクロンExpressα68.....98,000円
(SHARP X68000)

★CG大会には、αで応募しよう!!

お知らせ

サイクロンテクニカルセミナー in 大阪

大阪シャープROAショールームにて開催中。
お申し込みはアンスまで。

★ステップ3 7月26日(木)

「ポリゴンを使用する」Z² TRIPHONY DIGITAL CRAFTとのリンク
.....5,000円

★ステップ4 8月23日(木)

「絵を貼りつける」マッピングの使用法.....5,000円



株式会社アンス・コンサルタンツ

九州本社/〒810 福岡市中央区平丘町68
phone.092-522-6347 FAX092-521-0400

68000

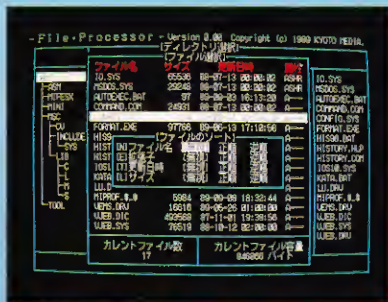
本格的ファイルマネージングソフトウェア

**業界の新星、ロゴスシステムが
ユーザーの希望を1つの形にしました。
これは必要だとか便利じゃない、快感だ!**

全国有名パソコンショップでお求め下さい。
電話1本での通信販売も受付いたしております。

THE FILE PROFESSORの実力

ディスクのバックアップ、ディスクのエディット、ディスクの初期化、ディスクの比較、ディスクの検査、ディスクの情報、FATのエディット、ファイルの検索、ディレクトリのコピー、ディレクトリの削除、ボリュームラベルの設定、ディレクトリの作成、ディレクトリ構造の再読み込み、ディレクトリ構造の印刷、ディレクトリ名の変更、ディレクトリ内容のソート、削除ファイルの復元、ファイル属性の変更、ファイルのコピー/移動、ファイルの削除、ファイルのエディット、ファイルの配置情報、ファイル一覧の印刷、ファイル名の変更、ファイルのソート、ファイル更新日時の変更、ファイルの表示、ファイルの発行、カレンダー、ハードディスクの直接エディット、システム情報の表示、コマンドシェル、現在時刻の変更。



**メニュー選択方式を実現!!
初心者でも簡単に使える**
(画面写真は、386用を開発中のものです。)

ロゴスシステム

このソフトはロゴスシステムのデビュー作です。でも、だからといってなめてもらっちゃあ困ります。私達は、いろいろなソフトを作りました。そのどれもが他社から発売されてきました。出来る事ならば自分達で発売したい/その願いがやっとかきました。

好評発売中!

ロゴスシステム

〒615 京都市右京区西院上今田町17-1 L&Pビル4F
TEL (075) 812-6383 FAX (075) 822-6915

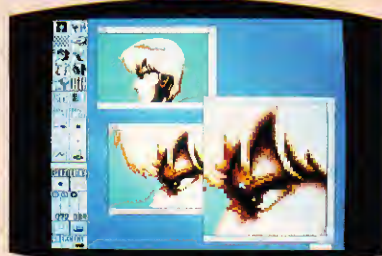
定価 **28,000円**

The File Professor

ザインの彩先端。 ニューコンセプトのアートキャンバス「G=ツール」、登場。

X68000ユーザーのクリエイティブマインドに火をつける新感覚のグラフィックツール。これまでのエディタ概念を払拭し、作品に挑むうえで必要不可欠なグラフィックキャラクター・背景作成のすべてを備えたトータルツールです。ゲームデザインをはじめとしたオリジナルコンピュータアートが驚くほど自由に描けます。今回はグラフィックやスプライトのキャラクターの作成を目的とした「GR EDITモード」をご紹介します。

GR EDITモード

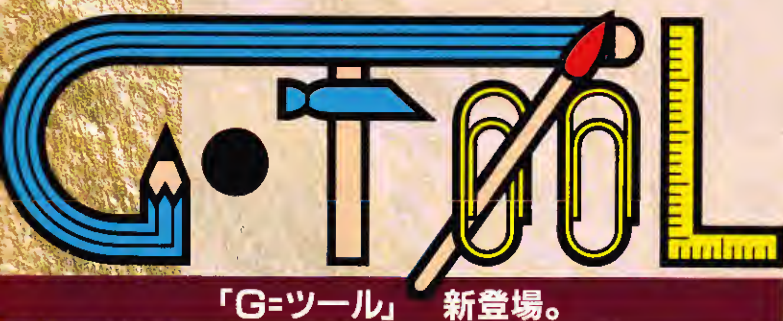
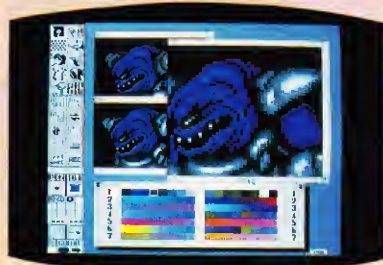


マルチウィンドウシステム：最大12枚まで描画ウィンドウが開ける優れたシステム環境を装備。複数のグラフィック・キャラクターが同時に作成できます。

ユーザーアイコンシステム：パレットやタイル、ペンなど、メインアイコン内の機能を使い勝手に合わせて、自分流のアイコン作成が可能。いちいちポップアップメニューを呼び出す必要もなくアートワークがはかどります。

マウス定義機能システム：マウスの左右クリックボタンに機能定義が可能。たとえば左利きの方もスムーズにオペレーティングできます。

高速メニューウィンドウ処理：メニューウィンドウの開閉は瞬時に。ユーザーアイコンシステムとの併用で、スピーディに仕事が進みます。



新発売
¥28,000

zainsoft

株式会社ザイン・ソフト
〒676兵庫県高砂市米田1162-1
TEL (0794) 31-7453

世にも楽しいシューティングパズル

クオース TM

QUARTH

©KONAMI 1990

**X68000版
好評発売中
6,800円(税別)**

MSX2版 好評発売中 5,800円(税別)

PC-9801版 近日発売予定

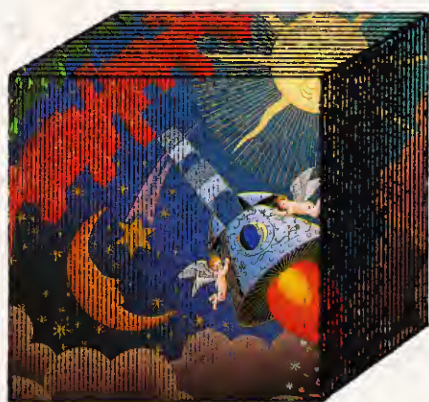
落ちて来るブロックを四角にして消してゆきます。一度にたくさん消すと効率的で得点も大幅アップ。下のラインまで来るとゲームオーバーです。





歴史に残る 前人未踏の 四角い宇宙

だれもが夢中になれるゲームを創りたい。シンプルでいて奥が深い。だれでも気軽に遊べて、いつまでも飽きない。そんなピュアな、ほんとうの意味でのゲームがしたいと思うことがある。ゲームに対する熱い想いをもう一度じっくりと見つめて今、コナミが新たに発進する、樂園ゲームプロジェクト「クオース」。シューティングの楽しさと、パズルの思考性がマッチングした、すでにゲームセンターでは爆発人気の極楽行き超ソフトだ。ほら、もう引力がココロをズルズルと吸いこんでいる。君も、友も、父も、母も、老若男女を巻きこんで、樂園へ行こう。



協力CPUで、仲直りもできる、ますます熱中の親切設計。



対戦CPUは、敵と相手の両方と戦う恐怖のケンカバトルだ。



アイテムブロックが出るにフッキー。

アーケード版
ジェミニウイング
待望の移植を実現！

ゲームセンターを賑わした
大人気シューティングゲーム
「ジェミニウイング」が、
キミのX68Kで今、蘇る！！

MIDI対応

ジェミニウイング Gemini WingTM

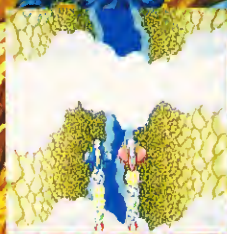
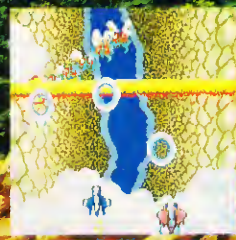
幾千の流星が降りそそいだ年、世界は蟲に覆われていた。人々は孤立し、街は滅び、植物に埋め尽くされた。蟲たちはさらに勢いを増し、残された僅かな地さえも蝕んでゆく。そして、ついに最高機密指令第307号、コード名ジェミニウイングは発動された……！

◆特徴◆

- 二人同時プレイ可能
- MIDI対応(※)
対応楽器 ローランド MT-32
CM-32L CM-64
(※)対応機種ごとに、それぞれ違ったBGMをお楽しみいただけます
(※初期のMT-32では正常に演奏できません)
- FM音源、ADPCM対応
- ジョイスティック対応
- 512HD 2枚組
- X68000 全シリーズ対応

標準価格 8,800円

Copyright ©1987TECMO



闇の血族

THE PREDESTINED HOMICIDES #1

美少女名探偵 魅由の繰り広げる
ミステリアスアニメーションアドベンチャー 第1弾!!

艶やかなファッション界を襲う奇怪な連続殺人事件。
南米の血に隠された秘密とは?
そして魅由を待ち受ける血族の宿命は?

NOVEL WARE

上
巻

あたし、魅由。

新宿にあるデザイン・スタジオの、新人A・D（アパレル・デザイナー）。……なんだけどあたしの持つてゐる妙な「力」みたいなモノ——人の心が判かっちゃったり、変にカンが良かったり——のせいで、周りからは「名探偵魅由」なんて呼ばれて、よく相談事を持ち込まれたりしている。で、そんなある日、友達のモデルが、突然、殺されてしまった。

そして、あたしの親友だった唯も……!

これって……ひょっとして連続殺人事件ってヤツ!?



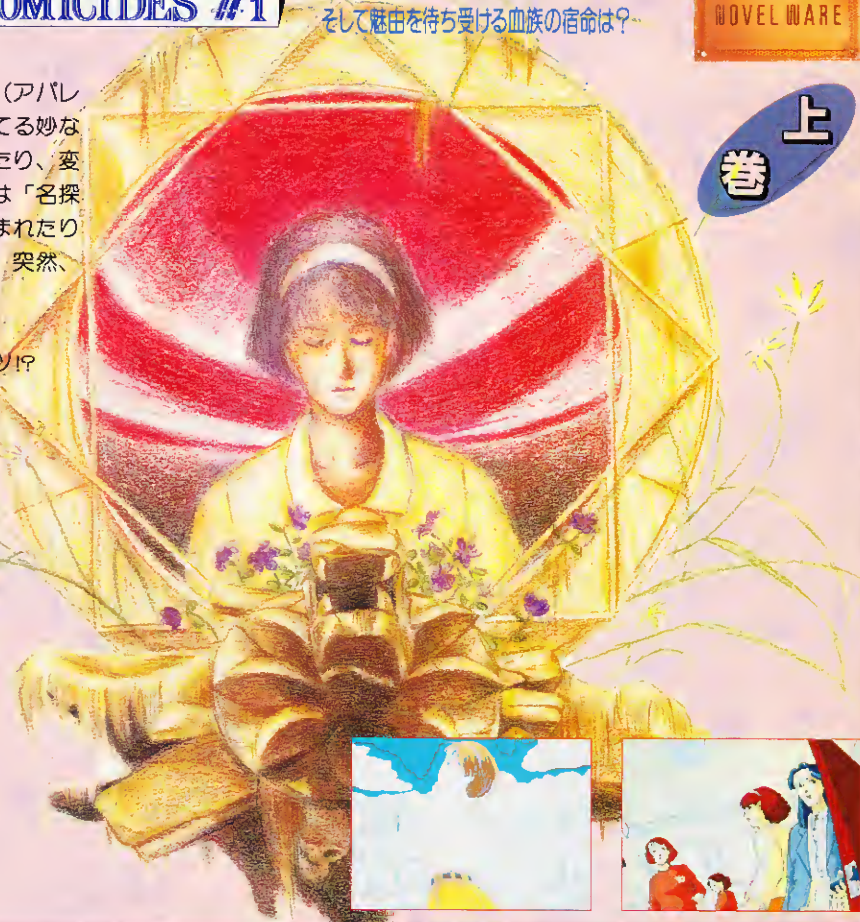
新発売!!



X68000対応 5"-2HD

●ローランド社MT32完全対応
MIDIインターフェイスボードC-Z-6BMI
又は、SACOM製SX-68Mが必要です。
(初期のMT-32では、正常に演奏できません。)

標準価格 8,800円



伊澤 魅由
(いざわ みゆ)

誕生日: 7月16日
身長: 168cm
体重: 57kg



姫野 里沙
(ひめの りさ)

誕生日: 4月2日
身長: 163cm
体重: 45kg



雪原 リーン
(ゆきはら リーン)

誕生日: 2月10日
身長: 170cm
体重: 53kg



高校生の時、デザイナーの泉麗子に見込まれ、学生生活を営む傍ら麗子のデザインスタジオ（専門学校）に通い始める。そこで小品の手伝いなどをしながら、デザイナーとして本格的に勉強を開始。2年間の研修期間を終え、高校卒業と同時に麗子の強力な推薦で、現在所属している〈スタジオY〇〉に入った。

〈スタジオY〇〉の専属モデル。ファッションショー、雑誌モデルを専門としている。平凡な可愛さがウリで、生活の中で“Y〇（自己性）”をファッションابلに演出する——といった〈スタジオY〇〉のメイン・コンセプトから考えれば、最もY〇らしいモデルと云えるかも知れない。

〈スタジオY〇〉の付属学校、「矢萩デザイナーズ・スタジオ」の卒業生。研修期間中「Y〇プロデュース」でスタイリスト補助のアルバイトをしていた。現在では、Y〇でファッションショーを中心とした若手スタイリストとして活躍中。

東芝EMIより
「38万キロの虚空」CD

新発売 MT税込価格 2,250円
CD税込価格 2,530円

ノベルウェアシリーズ
「38万キロの虚空」

PC-9801 X68000 FM-TOWNS
各9,800円

好評発売中!!
メタルサイト
×68000 8,800円

※標準価格には消費税は含まれておりません。



株式会社 システム サコム
〒130 東京都墨田区両国4-38-16
両国桜井ビル4F
ハードウェア部 03(635)5145
ソフトウェア部 03(635)7609

RPGの概念を一変させた傑作!

1988年発売と同時に世界中のゲーム・フリークを熱狂させた、あの「ダンジョン・マスター」が今、日本中を荒しまわる。

3Dグラフィックスによる複雑な迷路、数々の謎、パーティーを突然襲って来るモンスター。

そしてなによりもプレイヤーの考えること、見ること、手にすること、
すべてにリアルタイムで動いていく……本当の意味のリアルRPGだ。

なぜ、世界をそして日本をこれ程までに興奮させたのか!

その答えは君自身で出して欲しい。

Dungeon Master

ダンジョン・マスター

Master



※画面写真はX-68000版



好評発売中

■X68000
マウス対応

■PC-9801VM21/11, VX, RX, RS, RA ■PC-9800

■PC-9801UV21/11, UX, CV, EX, ES 要バス・マウス/アナログRGB対応

各¥9,800(税抜)

Produced by FTL Games © 1987, 1990 Software Heaven, Inc. © 1990 VICTOR MUSICAL INDUSTRIES, INC.



これが進化した麻雀ソフト、待望のX-68対応発売。

雀豪2

強
知
能
版

麻 雀ソフトの決定版登場! プレイすればするほど個性をもったプレイヤーに成長する
自己成長型サンプリング機能と、より強化された推論型人工知能の搭載で限りなく
実戦麻雀に近づいた。

リアルな4人囲みと見やすい麻雀牌、迫力ある効果音などの採用が麻雀ソフトの金字塔の
地位を不動のものにする。

■8月上旬発売:X-68000 ■好評発売中:PC-9801シリーズ

各¥9,800(税抜き)

※画面写真はX-68版の開発画面です

発売 ビクター音楽産業株式会社

通信販売

当社の商品をお近くのパソコンショップでお買い求めになれない場合、商品名、機種名、住所、氏名、電話番号を明記のうえ、下記住所まで
定価プラス3%消費税分を現金書留にてお申し込み下さい。(送料無料) 〒151 東京都渋谷区千駄ヶ谷2-8-16 ビクター音楽産業㈱(通信販売係)

X68000

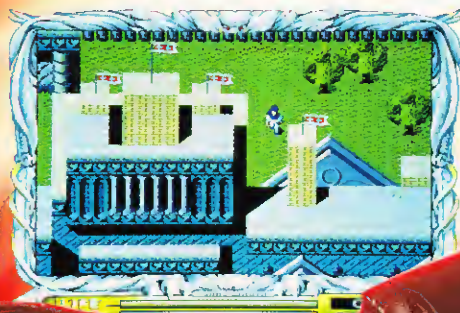
X68000の本質。「黒衣の貴公子」が今解き明かす。



Rune Worth is the world of the boundary between lightness and darkness.
Everything had been born there and then flourished and died there.

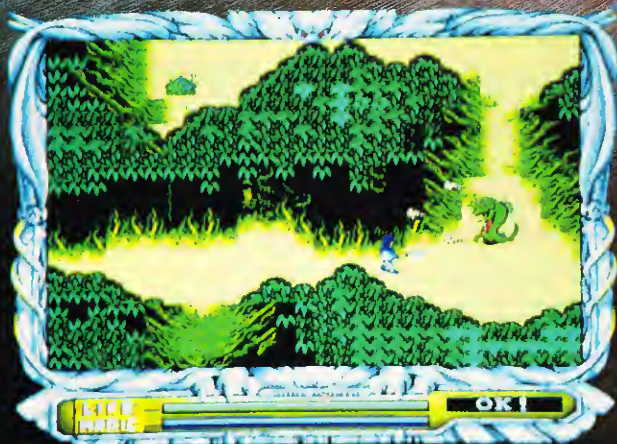
©1990 T & E SOFT

黒衣の貴公子



neXt

RPG・ACT・SLG. 最強のラインナップで
次世代体験…… neXt!



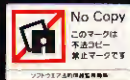
X68000版
7/13 FRI 新発売

- X68000 5"2HD 3枚組
- 全グラフィック書き起こし(高解像グラフィック 512×512ドット)
- ジョイスティック対応
- FM音源8音+ADPCM音源対応
- PC-9801 VM、UVシリーズ・PC-286、386シリーズ、NOTE対応
- 5"2HD/3.5"2HD 2枚組 ● サウンドボード対応 ● ジョイスティック対応
- ※ VM、UV RAM容量 640Kバイト以上必要です。
- ※ PC-386/1E/F/M/VF/U2/XAでは、ドライブ、RAM等の増設の如何にかかわらず、作動いたしません。
- ※ 1ドライブのみ搭載のPC-286L/LE/LFおよびPC-286NOTE Executiveでは、ドライブを増設しても作動いたしません。
- ※ 高解像度(640×480ドット)カラーディスプレイをお使いください。液晶ディスプレイにも対応しています。
- PC-8801 SRシリーズ・VA、98DO対応 5"2D 5枚組
- サウンドボード II 完全対応、ADPCMをフルサポート ● ジョイスティック対応(98DOを除く)
- NEC純正128KRAMボード、I/Oデータ機器製RAMボードに対応したキャッシュドライバ一搭載
- MSX2 MSX2+ (RAM64K以上、VRAM128K以上) 3.5"2DD 3枚組
- MSX-MUSIC(FM音源)対応 ● ジョイスティック対応

標準 各¥8,800

※表示価格に消費税は含まれません。

RPG-neXt……ルーンワース 黒衣の貴公子
ACT-neXt……幻 獣 鬼
SLG-neXt……遙かなるオーガスタ



- 通信販売ご希望の方は現金書留で料金と商品名・機種名と電話番号を明記の上、当社宛お送りください。(遠慮希望の方は300円プラス)
- カタログご希望の方は、送料として切手200円分を同封の上、カタログ請求券をお送りください。(兼書での請求はお断りします)
- T & Eの最新情報がわかるテレフォンサービス 名古屋(052)776-8500

T&E SOFT

企画・開発・製造・販売
株式会社 ティーアンドイーソフト

〒465 名古屋市長東区豊が丘1810番地 PHONE:052-773-7770

カタログ
請求券
Oh!x
8月号

注目!!

冬のボーナス一括払い
手数料(金利)無料

(平成2年12月末支払いをご利用下さい。)

モデム (AIWA) 50台限定 (送料¥1,000)
PV-A24MNP5 (定価 ¥54,800)
●MNPクラス5 限定特価¥26,500
●2400bps (送料・消費税込 ¥28,325)

CYBER STICK

●CZ-8NJ2
(定価 ¥23,800)
超特価!!



¥18,500 (送料・消費税込 ¥19,570)

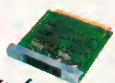
X68000シリーズ専用 特価¥16,480

MIDIインターフェースボード

SX-68M (サコム)

(純生コンパ) 定価 ¥19,800

送料・消費税込み!



またまた

秋葉原でおなじみ

7/15~8/15

X-1ターボZIII 特別ご提供品!!

台数限定

●CZ-888C+CZ-860D+M-2HD (10枚)

定価 ¥269,600▶特価¥164,800

(ボーナス併用も有りますTEL下さい)

●ジョイカード
●ゲーム3種
●パソコンラック A 3段
プレゼント中
送料消費税込み!!

12回 14,400 24回 7,600 36回 5,300 48回 4,100 60回 3,400

●お近くの方は

●本体単品で

●ビジネスソフト

ジョイスティック 送料 ¥500

●X-1PRO

定価 ¥9,500▶特価 ¥7,800

●ASCII STICK

定価 ¥6,800▶特価 ¥5,500

NEW X68000EXPERT II/II-HD & PROII/PROII-HD & SUPER-HD

(送料・消費税込)

EXPERT II

セットでお買い上げの方に、

- ディスク10枚
- ゲーム3種
- ジョイカード2枚

プレゼント中!!

EXPERT II-HD

セットでお買い上げの方に、

- ディスク10枚
- ゲーム3種
- ジョイカード2枚

プレゼント中!!

PROII

セットでお買い上げの方に、

- ディスク10枚
- ゲーム3種
- ジョイカード2枚

プレゼント中!!

PROII-HD

セットでお買い上げの方に、

- ディスク10枚
- ゲーム3種
- ジョイカード2枚

プレゼント中!!

SUPER-HD

セットでお買い上げの方に、

- ディスク10枚
- ゲーム3種
- ジョイカード2枚

プレゼント中!!

EXPERT II

A) セット: CZ-603C+CZ-604D	定価 ¥432,800▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	
B) セット: CZ-603C+CZ-605D	定価 ¥453,000▶特価 (価格はお電話下さい)
12回 30,200 24回 15,900 36回 11,000 48回 8,500 60回 7,100	
C) セット: CZ-603C+CZ-613D	定価 ¥473,000▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	
D) セット: CZ-603C+CU-21HD	定価 ¥466,000▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	

EXPERT II-HD

A) セット: CZ-613C+CZ-604D	定価 ¥542,800▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	
B) セット: CZ-613C+CZ-605D	定価 ¥563,000▶特価 (価格はお電話下さい)
12回 37,700 24回 19,800 36回 13,700 48回 10,600 60回 8,900	
C) セット: CZ-613C+CZ-613D	定価 ¥583,000▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	
D) セット: CZ-613C+CU-21HD	定価 ¥596,000▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	

PROII

A) セット: CZ-653C+CZ-604D	定価 ¥379,800▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	
B) セット: CZ-653C+CZ-605D	定価 ¥400,000▶特価 (価格はお電話下さい)
12回 26,800 24回 14,100 36回 9,700 48回 7,600 60回 6,300	
C) セット: CZ-653C+CZ-613D	定価 ¥420,000▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	
D) セット: CZ-653C+CU-21HD	定価 ¥433,000▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	

PROII-HD

A) セット: CZ-663C+CZ-604D	定価 ¥489,800▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	
B) セット: CZ-663C+CZ-605D	定価 ¥510,000▶特価 (価格はお電話下さい)
12回 34,100 24回 17,900 36回 12,400 48回 9,600 60回 8,100	
C) セット: CZ-663C+CZ-613D	定価 ¥530,000▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	
D) セット: CZ-663C+CU-21HD	定価 ¥543,000▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	

SUPER-HD

A) セット: CZ-623TN+CZ-604D	定価 ¥592,800▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	
B) セット: CZ-623TN+CZ-605D	定価 ¥613,000▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	
C) セット: CZ-623TN+CZ-613D	定価 ¥633,000▶特価 (価格はお電話下さい)
12回 42,700 24回 22,500 36回 15,500 48回 12,100 60回 10,100	
D) セット: CZ-623TN+CU-21HD	定価 ¥646,000▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	

X68000シリーズ ~P&Aスペシャルセット=限定誌上販売!!



台数限定

送料、消費税込み

セットでお買い上げの方に、
●ディスク10枚 ●ゲーム3種 ●ジョイカード2個 プレゼント中

EXPERT

- CZ-602C+CZ-612D 定価 ¥475,800▶特価¥306,000
- CZ-602C+CZ-604D 定価 ¥450,800▶特価¥300,000
- CZ-602C+CZ-605D 定価 ¥471,000▶特価¥320,000
- CZ-602C+CZ-613D 定価 ¥491,000▶特価¥336,000
- CZ-602C+CU-21HD 定価 ¥504,000▶特価¥338,000

EXPERT-HD

- CZ-612C+CZ-612D 定価 ¥585,800▶特価¥375,000
- CZ-612C+CZ-604D 定価 ¥560,800▶特価¥369,000
- CZ-612C+CZ-605D 定価 ¥581,000▶特価¥386,000
- CZ-612C+CZ-613D 定価 ¥601,000▶特価¥403,000
- CZ-612C+CU-21HD 定価 ¥614,000▶特価¥407,000

PRO-HD

- CZ-662C+CZ-612D 定価 ¥527,800▶特価¥339,000
- CZ-662C+CZ-604D 定価 ¥502,800▶特価¥333,000
- CZ-662C+CZ-605D 定価 ¥523,000▶特価¥352,000
- CZ-662C+CZ-613D 定価 ¥543,000▶特価¥368,000
- CZ-662C+CU-21HD 定価 ¥556,000▶特価¥372,000



1回～84回払いまでOK!!

★頭金なし!★即日発送

P&Aがズバリ超特価セールでご奉仕!!

●価格は流通事情により変動致しますので、銀行振込・書留等の送付前に、あらかじめお電話にてご確認ください。

立寄り下さい。専門係員が説明いたします。
 価で受付します。詳しくは電話にてお問合せ下さい。
 価の20%引きOK! TELください。

全国通販

X68000用ソフトコーナー (送料1ヶ～5ヶまで¥500)

Z's STAFF PRO68K Ver.2.0(ツァイト)	定価 ¥ 58,000	特価 ¥ 39,700
Z's TRIPHONY デジタルクラフト(ツァイト)	定価 ¥ 39,800	特価 ¥ 29,300
テラツォ(ハンダバード)	定価 ¥ 19,800	特価 ¥ 15,800
KAMIKAZE (サムシング・ウッド)	定価 ¥ 68,800	特価 ¥ 46,000
EW & EI(イースト)	定価 ¥ 38,800	特価 ¥ 28,800
G&P Professional Pack (マイクロウェアジャパン)	定価 ¥ 58,800	特価 ¥ 43,000
Final Ver.3.2(エーエスピー)	定価 ¥ 38,000	特価 ¥ 30,000
DATA PRO68K CZ220BS	定価 ¥ 58,000	特価 ¥ P&A特価
CARD PRO68K CZ226BS	定価 ¥ 29,800	TEL下さい!
C compiler PRO68K CZ211LS	定価 ¥ 39,800	特価 ¥ 32,000
OS-9/X68000 CZ219SS	定価 ¥ 29,800	特価 ¥ P&A特価 TEL下さい!
AI-68K CZ234LS	定価 ¥ 188,000	特価 ¥ 143,000
THE 橋渡 V2.0 CZ224LS	定価 ¥ 9,900	特価 ¥ 7,700
SOUND PRO68K	定価 ¥ 15,800	特価 ¥ 12,500
MUSIC PRO68K CZ213MS	定価 ¥ 15,800	特価 ¥ P&A特価 TEL下さい!
Sampling PRO68K CZ215MS	定価 ¥ 17,800	特価 ¥ 14,000
MUSIC-studio PRO68K 237MS	定価 ¥ 15,800	特価 ¥ P&A特価 TEL下さい!
MUSIC-PRO68K (MIDI) 247MS	定価 ¥ 28,800	特価 ¥ 22,000
Newprint Shop 221HS	定価 ¥ 19,800	特価 ¥ P&A特価
Communication 223CS	定価 ¥ 19,800	TEL下さい!
O-TRACE68 Ver.3.0(キャスト)	定価 ¥ 98,000	特価 ¥ 77,000
サイクロンEXPRESS α68	定価 ¥ 98,000	特価 ¥ 72,000
68K Ver.2 PRO	定価 ¥ 22,000	特価 ¥ 16,300
THE FILE PRO68K (ロゴシステム)	定価 ¥ 28,000	特価 ¥ 20,500
ツクール(サンソフト)	定価 ¥ 28,000	特価 ¥ 20,500
たーみのる2 (SPS)	定価 ¥ 17,800	特価 ¥ 13,500
マジックバレット(ミュージカルプラン)	定価 ¥ 19,800	特価 ¥ 14,900
Hyper word OZ-251BS	定価 ¥ 39,800	特価 ¥ 30,900

●ゲームソフト 20%OFF OK!! (一部ソフト除く)

周辺機器コーナー (送料 ¥1,000)

A CZ-8NSI	定価 ¥ 188,000	特価 ¥ 145,000
B CZ-6VTI	定価 ¥ 69,800	特価 ¥ 54,000
C CZ-6TU	定価 ¥ 33,100	特価 ¥ 25,000
D BF-68PRO	定価 ¥ 19,800	特価 ¥ 15,500
E CZ-6BEI	定価 ¥ 35,000	特価 ¥ 26,500
F CZ-6BE1A	定価 ¥ 38,000	特価 ¥ 28,600
G CZ-6BE2	定価 ¥ 79,800	特価 ¥ 60,000
H CZ-6BE4	定価 ¥ 138,000	特価 ¥ 107,000
I CZ-6BFI	定価 ¥ 49,800	特価 ¥ 38,200
J CZ-6BPI	定価 ¥ 79,800	特価 ¥ 61,000
K CZ-6BMI	定価 ¥ 25,800	特価 ¥ 20,300
L CZ-6EBI	定価 ¥ 88,000	特価 ¥ 67,500
MAN-S100	定価 ¥ 36,600	特価 ¥ 28,500
N CZ-6SDI	定価 ¥ 44,800	特価 ¥ 35,000
O CZ-8PC3	定価 ¥ 65,800	
P CZ-8PC4	定価 ¥ 99,800	
Q CZ-8PG1	定価 ¥ 130,000	
R CZ-8PG2	定価 ¥ 160,000	
S CZ-8PKI	定価 ¥ 97,800	
T CZ-8PVI	定価 ¥ 198,000	特価 ¥ 153,000
U IO-735X	定価 ¥ 248,000	特価 ¥ 190,000
V CZ-8BSI	定価 ¥ 23,800	特価 ¥ 19,000
W PIO-6BE1-A(I/O DATA)	定価 ¥ 25,000	特価 ¥ 18,200
X PIO-6BE2-2M(I/O DATA)	定価 ¥ 50,000	特価 ¥ 36,800
Y PIO-6BE4-4M(I/O DATA)	定価 ¥ 88,000	特価 ¥ 64,800

P&A超特価 TEL下さい!

X68000用ハードディスク (送料 ¥1,000)

アイテム	
●HXD-040(40MB/23ms)	定価 ¥118,000 ▶ 特価 ¥ 88,000
●HXD-042(増設用)	定価 ¥128,000 ▶ 特価 ¥ 95,000
アイテック	
●ITX-640(40MB/28ms)	定価 ¥158,000 ▶ 特価 ¥101,000
●ITX-680(80MB/20ms)	定価 ¥198,000 ▶ 特価 ¥131,000

プリンター(ケーブル・用紙付)限定5台 新品(送料 ¥1,000)

●CZ-8PC3(カラー漢字24ドット熱転写プリンター)	定価 ¥65,800	特価 ¥39,800
●CZ-8PK8(24ピン漢字プリンター136桁)	定価 ¥152,000	特価 ¥69,000
●CZ-8PC4 P&A特選!!(カラー漢字48ドット熱転写プリンター)	定価 ¥99,800	特価 ¥56,000

モデムコーナー (送料 ¥1,000)

A) MD-24FS5(オムロン)	定価 ¥ 49,800	特価 ¥ 34,800
B) MD-24FS7(オムロン)	定価 ¥ 64,800	特価 ¥ 45,000
C) コムスター2424/4(NEC)	定価 ¥ 38,800	特価 ¥ 28,000
D) コムスター2424/5(NEC)	定価 ¥ 44,800	特価 ¥ 32,000

P & A 特選パソコンラック (送料無料) 移動自由(キャスター付)

① 3段 875(H) ×580(D) ×610(W) ¥9,000	② 4段 1320(H) ×600(D) ×630(W) ¥12,000	③ 5段 1280(H) ×600(D) ×620(W) ¥15,000
--	--	--

中古パソコン 送料 ¥2,000

●X-68000セット	▶ ¥210,000	●CZ-856C	▶ ¥45,000	●CU-14AG2	▶ ¥30,000
●X-68000ACEセット	▶ ¥240,000	●CZ-870C	▶ ¥55,000	●CU-14H2	▶ ¥30,000
●X-1ターボZセット	▶ ¥100,000	●CZ-881C	▶ ¥65,000	●CZ-8PC2	▶ ¥25,000
●X-1G/30セット	▶ ¥39,000	●CZ-820D	▶ ¥10,000	●CZ-8PK6	▶ ¥32,000
●CZ-822C	▶ ¥15,000	●CU-14GB	▶ ¥5,000		
●CZ-830C	▶ ¥25,000	●CU-14BD	▶ ¥25,000		

通信販売お申し込みのご案内

[現金一括でお申し込みの方]

●商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで、現金書留でお送りください。(プリンター・フロッピーの場合、本体使用機種名を明記のこと)

[銀行振込でお申し込みの方]

●銀行振込ご希望の方は必ずお振込みの前にお電話にてお客様のご住所・お名前・商品名等をお知らせください。

(電信扱いでお振込み下さい)

[振込先] 住友銀行 新小岩支店
当No.263914 (株)ピー・アンド・エー

[クレジットでお申し込みの方]

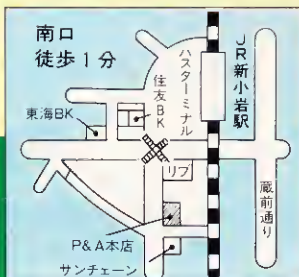
●電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。

●現金特別価格でクレジットが利用できます。残金のみに金利がかかります。

●1回～84回払いまで出来ます。但し、1回のお支払い額は¥1000円以上。

超低金利クレジット率

回数	3	6	10	12	18	24	36	48	60	72	84
手数料	2.5	3.5	5.0	5.0	9.0	10.5	14.5	19.0	24.5	32.0	38.5



中古パソコンはP&Aにおまかせ!!

●まずはお電話下さい。 ●下取り・買取りでお急ぎの方、直接当社に来店、または、宅急便にてお送り下さい。

●下取りの場合..... 価格は常に変動していますので査定額をお電話で確認して下さい。(差額は、P&A超低金利クレジットをご利用下さい。)

●買取りの場合..... 現品が着次第、2日以内に買取り金額を連絡し、振込み、又は書留でお送り致します。

●近郊の方は、P&A本店まで、直接お持ち下さい。即金にて、¥1,000,000までお支払い致します。

《便利な超低金利クレジットをご利用下さい》

- 月々¥1,000円からOK!! ●ボーナス払いOK(夏冬10回までOK)
- 支払い回数 1回～84回 ●お支払いは、8ヶ月先からでもOK!!

アフターサービス万全

全商品保証付。専門の担当者がお客様の立場で対応します。
 初期不良、輸送トラブル等。
 万が一初期不良、輸送トラブルが発生した際には、即交換させていただきます。

●定休日/毎週水曜日=第3水曜・木曜は連休とさせていただきます(祭日の場合は翌日になります)

- マイコン
- ビデオ
- ビデオテープ

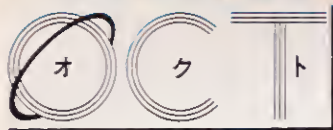
P&A

株式会社ピー・アンド・エー
 〒124 東京都葛飾区新小岩2丁目1番地19号

☎03-651-0148(代) FAX. 03-651-0141

営業時間
 平日:AM10:00～PM7:00
 日祭:AM10:00～PM6:00

●現金書留及び銀行振込でお申し込みの方は、上記商品の料金の3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合わせ下さい。



03-730-6271

●営業時間 AM 11:00 ~ 9:00/日曜・祭日PM7:00 電話一本で、ハイ即納
〒144 東京都大田区蒲田4-6-7 FAX 03-730-6273

●定休日毎週火曜日 祭日の場合翌日になります。
オクト
全国通販
ラクらくクレジット

OCT-1 システム インフォメーション

- ▶全商品保証付(メーカー保証)
- ▶超低金利ハッピークレジット(1回~60回)頭金ナシOK!
- ▶ボーナス一括払いOK/ボーナス2回払いOK!
- ▶配達日の指定OK/(万全なサポート体制)
- ▶商品の組合せ自由/オクトフリーダムシステム
- ▶店頭デモンストレーション実施中

オクト
セレクトッドシステム

広告掲載商品以外の
製品も取扱っております。



案内図



店頭セール実施中

OCT-1

蒲田

●平成2年、8月末一括払い(手数料ナシ)OK!!
OKだよ~ん。超低金利 ハッピークレジットですヨ
EXPERT II・PRO II 新発売記念セール開催中!!

OPEN

★下記セットでお買い上げの方にはプレゼント!! ●①MD-2HD 10枚 ②ジョイカード 2個 (連射式) ③シリコンキーボードカバー

お好みのセットをお選び下さい。15型カラーディスプレイTV

- SX-WINDOW搭載。
- 40Mバイトハードディスク搭載

送料無料

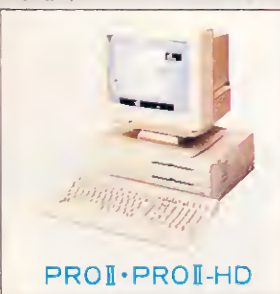


EXPERT II・EXPERT II-HD

- CZ-603C-BK/GY
定価 ¥ 338,000
- CZ-613C-BK/GY
定価 ¥ 448,000

現金特価!! 推選
お電話下さい。

- SX-WINDOW搭載。
- 拡張I/Oポート4スロット装備



PRO II・PRO II-HD

- CZ-653C-BK/GY
定価 ¥ 285,000
- CZ-663C-BK/GY
定価 ¥ 395,000

CZ-8NJ2

- インテリジェントコントローラ

定価 ¥ 23,800
超特価 ¥ 18,800



CZ-605D-GY/BK
定価 ¥ 115,000

15型カラーディスプレイTV



CZ-613D-GY/BK
定価 ¥ 135,000

14型カラーディスプレイ



CZ-604D-GY/BK
定価 ¥ 94,800

21型カラーディスプレイ



CU-21HD
定価 ¥ 148,000

A CZ-603C + CZ-605D 定価合計 ¥ 453,000 ▶ オクト大特価

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

B CZ-613C + CZ-605D 定価合計 ¥ 563,000 ▶ オクト大特価

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

C CZ-653C + CZ-605D 定価合計 ¥ 400,000 ▶ オクト大特価

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

D CZ-663C + CZ-605D 定価合計 ¥ 510,000 ▶ オクト大特価

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

E CZ-603C + CZ-613D 定価合計 ¥ 473,000 ▶ オクト大特価

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

F CZ-613C + CZ-613D 定価合計 ¥ 583,000 ▶ オクト大特価

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

G CZ-653C + CZ-613D 定価合計 ¥ 420,000 ▶ オクト大特価

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

H CZ-663C + CZ-613D 定価合計 ¥ 530,000 ▶ オクト大特価

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

I CZ-603C + CZ-604D 定価合計 ¥ 429,800 ▶ オクト大特価

12回	¥ 28,000	24回	¥ 14,800	36回	¥ 10,200	48回	¥ 8,000
-----	----------	-----	----------	-----	----------	-----	---------

J CZ-613C + CZ-604D 定価合計 ¥ 542,000 ▶ オクト大特価

12回	¥ 36,000	24回	¥ 19,000	36回	¥ 13,100	48回	¥ 10,200
-----	----------	-----	----------	-----	----------	-----	----------

K CZ-653C + CZ-604D 定価合計 ¥ 379,800 ▶ オクト大特価

12回	¥ 25,400	24回	¥ 13,400	36回	¥ 9,300	48回	¥ 7,200
-----	----------	-----	----------	-----	---------	-----	---------

L CZ-663C + CZ-604D 定価合計 ¥ 489,800 ▶ オクト大特価

12回	¥ 32,200	24回	¥ 17,000	36回	¥ 11,800	48回	¥ 9,200
-----	----------	-----	----------	-----	----------	-----	---------

M CZ-603C + CU-21HD 定価合計 ¥ 486,000 ▶ オクト大特価

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

N CZ-613C + CU-21HD 定価合計 ¥ 596,000 ▶ オクト大特価

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

O CZ-653C + CU-21HD 定価合計 ¥ 433,000 ▶ オクト大特価

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

P CZ-663C + CU-21HD 定価合計 ¥ 543,000 ▶ オクト大特価

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

♡どんどんTELしよう。安くなるかもヨ!!

♡クレジット価格は、消費税込みですヨ。ご利用下さい!!

※クレジットの回数は1回~60回、ボーナス併用などありますのでお電話でお問合せ下さい。

■本体セット:送料無料 ●店頭デモ実施中...専門の係員が詳細にアドバイス致します。ぜひご来店下さい。

※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは、電話でお問合せ下さい。

平成2年夏のボーナス一括払い(8月末)OK!!手数料ナシ!!おトクです。ぜひ!!超低金利クレジットをご利用下さい。

■店頭にて、ゲームソフト25%OFF!! (税別) / 超低金利クレジットをご利用ください!!
■特に人気のある商品によっては、しばらくお待ち願うことがありますのでご了承下さい。

厳選された製品を、より安く、より早く、皆様のお手元に!!

広告掲載商品以外の
製品も取扱っております。

チャンス / X68000・SUPER-HD(チタン) = 好評・発売中
とんとんTEL下さいネ。 送料 ¥2,000

SX-WINDOW搭載。



●ザ・ワークステーションと呼ぶにふさわしい
スーパーな68000!! 新登場!!
SUPER-HD。

※プレゼント / ①MD-2HD10枚 ③ジョイカード(連射式)
②アフターバーナー(¥9,200) ④シリコンキーボード(¥2,800)

X68000 SUPER-HD

●CZ-623C-TN+CZ-613D-TN
定価合計 ¥633,000... 大特価!! TEL下さい。

※マウス・トラックボール付!! ディスプレイにはスピーカ2個、チルト台付!!

12回 ? 24回 ? 36回 ? 48回 ?

♡安くてゴメンなさい。今だけヨ!!

他のディスプレイ: ①CZ-602D、②612D、③CZ-603D、
④CU-21HDの組合せもごさいますのでお問い合わせ下さい。

※超低金利クレジットご利用下さい。1回~60回払い、頭金ナシ / ボーナス1回払い、ボーナス2回払いOK!!

X68000 EXPERT-HD

オクト限定スペシャルセット



ラストチャンス!!
早い者勝ち!!

- CZ-612C(BK) (¥466,000)
- CZ-602D(BK) (¥99,800)
- MD-2HD 10枚
- ジョイカード(連射式×2個)
- ゲーム

オクト超特価
¥364,000 (送料・消費税込み!!)

※ディスプレイ=①CZ-604D ②CZ-605D
③CZ-613D ④CU-21HD
との組合せもごさいます。TEL下さい。

オクト特選 シャープ周辺機器 (送料 ¥1,000)

- CZ-6BE1 IBM増設RAMボード (¥35,000) ▶特価 ¥26,500
- CZ-6BE1B IBM増設RAMボード (¥28,000) ▶特価 ¥21,000
- CZ-6BE2 2MB増設RAMボード (¥79,800) ▶特価 ¥60,500
- CZ-6BE4 4MB増設RAMボード (¥138,000) ▶特価 ¥104,800
- CZ-6BF1 増設用RS-232Cボード (¥49,800) ▶特価 ¥38,500
- CZ-6BG1 GPIBボード (¥59,800) ▶特価 ¥45,000
- CZ-6BM1 MIDIボード (¥26,800) ▶特価 ¥20,500
- CZ-6BN1 スキャナ用ハラルボード (¥29,800) ▶特価 ¥22,800
- CZ-6BP1 数値演算プロセッサボード (¥79,800) ▶特価 ¥60,500
- CZ-6BO1 ユニバーサルI/Oボード (¥39,800) ▶特価 ¥30,500
- CZ-6EB1/BK 拡張I/Oボックス (¥88,000) ▶特価 ¥66,800
- CZ-6VT1/BK カラーイメージユニット (¥69,800) ▶特価 ¥53,000
- CZ-6BL1 LANボード (¥268,000) ▶大特価

- CZ-8NM2A マウス (¥68,800) ▶特価 ¥5,300
- CZ-8NT1 マウス・トラックボール (¥98,800) ▶特価 ¥7,500
- CZ-8NS1 カラーイメージスキャナ (¥188,000) ▶大特価
- CZ-8BG1 FAXボード (¥79,800) ▶特価 ¥60,500
- CZ-8TM2 モデムユニット (¥49,800) ▶特価 ¥38,000
- CZ-64H 増設ハードディスク (¥120,000) ▶大特価
- CZ-6TU GY/BK RGBシステムチューナー (¥33,100) ▶特価 ¥25,000
- BF-68PRO 高性能CRTフィルター (¥19,800) ▶特価 ¥15,500
- SX-68M(システムスコム) MIDIボード (¥19,800) ▶特価 ¥15,000
- PIO-68BE1A(I/O DATA) IBM増設RAMボード (¥25,000) ▶特価 ¥18,500
- PIO-68E2-2M(I/O DATA) 2MB増設RAMボード (¥50,000) ▶特価 ¥37,000
- PIO-68E4-4M(I/O DATA) 3MB増設RAMボード (¥88,000) ▶特価 ¥65,000

オクト面白グッズ

アイテック(送料 ¥1,000)

- IT-X640 (¥158,000) ▶特価 ¥103,000
- IT-X680 (¥198,000) ▶特価 ¥134,000

モデムコーナー(送料 ¥1,000)

- MD-1200A III (送料 ¥1,000) ▶特価 ¥14,800
- MD-24FS4 (送料 ¥1,000) ▶特価 ¥31,500
- MD-24FS5 (送料 ¥1,000) ▶特価 ¥34,800
- MD-24FP4 (送料 ¥1,000) ▶特価 ¥27,900
- MD-12FS (送料 ¥1,000) ▶特価 ¥15,000

熱転写カラー漢字プリンター (クーポン紙付 送料 ¥1,000)

CZ-8PC4 ¥99,800

限定

●48ドット

サーマルヘッド

●B5~B4まで

●ハガキ可能

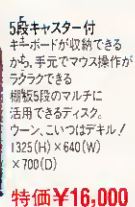
●カラー対応

オクト大特価 ¥55,800

- ①CZ-8PG3(24ドット熱転写カラー漢字プリンター)
定価 ¥65,800... 特価 ¥45,000
- ②CZ-8PK9(24ピン漢字プリンター80桁)
定価 ¥89,800... 大特価!! TEL下さい。
- ③CZ-8PK10(24ピン漢字プリンター136桁)
定価 ¥97,800... 大特価!! TEL下さい。
- ④CZ-8PG1(24ピンカラー漢字プリンター80桁)
定価 ¥130,000... 大特価!! TEL下さい。
- ⑤CZ-8PG2(24ピンカラー漢字プリンター136桁)
定価 ¥160,000... 大特価!! TEL下さい。
- ⑥IO-735x(カラーイメージジェット)
定価 ¥248,000... 大特価!! TEL下さい。

パソコンラック 推奨 送料 無料

①五段キャスター付 ②四段キャスター付 ③三段キャスター付



5段キャスター付
キーボードが収納できる
から、手でマウス操作が
ラクができる
棚板5段のマルチに
活用できるデスク。
ウーン、こいつはデキル!
1325(H)×640(W)
×700(D)

4段キャスター付
どんなパソコンにも
フレキシブルに対応!
使い易いデスクです。
1245(H)×614(W)
×600(D)

3段キャスター付
場所を選ばない
簡易で便利な
デスクです。
1175(H)×640(W)
×600(D)

特価 ¥16,000

特価 ¥12,000

特価 ¥8,800

X68000ソフト大セール実施中※ゲームソフト25%off

型名	商品	定価	特価
QZ-211LS	Complier PRO-68K	¥39,800	¥28,800
QZ-212BS	BUSINESS PRO-68K	¥68,000	¥48,000
QZ-213MS	MUSIC PRO-68K	¥18,800	¥13,500
QZ-214MS	SOUND PRO-68K	¥15,800	¥11,500
QZ-215MS	Sampling PRO-68K	¥17,800	¥12,800
QZ-219SS	OS-9 X68000	¥29,800	¥21,000
QZ-220BS	DATA PRO-68K	¥58,000	¥41,000
QZ-221HS	New Print Shop PRO-68K	¥19,800	¥14,300
QZ-223CS	Communication PRO-68K	¥19,800	¥14,300
QZ-224LS	THE 福袋 V2.0	¥9,900	¥7,500
QZ-226BS	CARD PRO-68K	¥29,800	¥21,300
QZ-241BS	システム手帳リファレンス	¥9,800	¥7,500
QZ-242BS	活用フォーム集	¥9,800	¥7,500
QZ-244SS	Human 68K Ver.2.0	¥9,800	¥7,500
QZ-247MS	MUSIC PRO-68K(MIDI)	¥28,800	¥20,800
QZ-240BS	Stationery PRO-68K	¥14,800	¥11,500
QZ-243BS	CYBER NOTE PRO-68K	¥19,800	¥15,200
EW		¥38,000	¥29,800
G-68K		¥14,800	¥11,400
E-68		¥19,800	¥15,300

★オクト今月だけの新品限定販売(各1台限)(送料 ¥1,000)

- CZ-822C(BK) 定価 ¥ ? 大特価 ¥18,800
- CZ-888C(BK) 定価 ¥168,000 大特価 ¥69,800
- CZ-601C(BK) 定価 ¥319,800 大特価 ¥174,000
- CZ-611C(BK) 定価 ¥399,800 大特価 ¥218,000
- CZ-652C(BK) 定価 ¥298,000 大特価 ¥188,000
- CZ-662C(BK) 定価 ¥408,000 大特価 ¥248,000
- CZ-601D(BK) 定価 ¥119,800 大特価 ¥68,000
- CZ-601D(GY) 定価 ¥119,800 大特価 ¥68,000
- CZ-612D(GY) 定価 ¥119,800 大特価 ¥74,000

店頭ゲームソフト25%off! ビジネスソフト 25%より特価中

★通信販売お申込みのご案内★ 〒144 東京都大田区蒲田4-6-7 TEL: 03-730-6271

お申込みはお電話でお願いします。お客様の住所・氏名・電話番号及び商品名をお知らせ下さい。入金確認後、商品をお送りいたします。

現金払い

銀行振込: お近くの銀行より(電信扱い)にて
お振込み下さい。
現金書留: 封筒の中に住所・氏名・商品名を
ご記入の上当社までお送り下さい。

クレジット

専用お申込用紙をお送り致します。
ので、必要事項をご記入、ご捺印の上
ご返送下さい。手続きは簡単です。

オクト ラクラク クレジット表

1回	2%	3回	2.5%	6回	3.5%	10回	5%
12回	5%	15回	7.5%	18回	9%	20回	10%
24回	11%	30回	14.5%	36回	15.5%	48回	20%

振込先

富士銀行 三菱銀行
久ヶ原支店 蒲田支店
⑤No.1824 ⑤No.0278691
株式会社 億人(オクト)

※掲載の価格は変動しますので、まずは、お電話にてご確認ください。

※連休のお知らせ=7/31(水)、8/1(水)は連休です。

※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは電話でお問合せ下さい。

※銀行振込、または、現金書留でご注文の際には、あらかじめ電話でご確認の上、お申し込み下さい。

平成2年夏のボーナス一括払いOK!! (8月末)手数料ナシ!!

超低金利クレジットをご利用下さい。

夏ツクモ! ガバーデン! 決算セール

掲載商品2万円以上
送料無料!!
は7/31(火)迄です。

ツクモ決算! 展示棚ズレ品

SHARP PA-6500
定価 ¥17,800
限定3台
55% OFF

決算特価 ¥9,800

SHARP PA-7000
定価 ¥19,800
限定3台
51% OFF

決算特価 ¥9,800

SHARP CZ-8PC3
定価 ¥65,800 限定3台
24ドット熱転写カラー複写プリンター
60% OFF

決算特価 ¥39,800

SHARP CZ-8PK7
定価 ¥122,000
24ピン、80桁
限定5台
51% OFF

決算特価 ¥59,800

SHARP CZ-8PK8
定価 ¥152,000
24ピン、136桁
限定5台
45% OFF

決算特価 ¥83,800

ツクモ通販受注センターフリーダイヤル

0120(377)999

商品のお問い合わせは各店又は通販部 ☎03(251)9911へ

LET'S MUSIC

Aセット

CM-32L ¥69,000
SX-68M ¥19,800
Musicstudio Mu-1 ¥19,800

合計定価 ¥108,600

ツクモ特価 ¥91,800

(消費税別途 ¥2,754)

クレジット例(税込)月々 ¥5,830 × 18回払

★Musicstudio PRO-68K V1.1又は、Music PRO68K(MIDI)のソフトの場合には ¥8,900プラスになります。

Bセット

CM-64 ¥129,000
SX-68M ¥19,800
Musicstudio Mu-1 ¥19,800

合計定価 ¥168,600

ツクモ特価 ¥144,000

(消費税別途 ¥4,320)

クレジット例(税込)月々 ¥7,107 × 24回払

電子手帳 & ポケコン

PA-8600 特価 ¥24,800

PA-7500 特価 ¥17,800

PC-E500 特価 ¥24,800



情報ツール

All in Note



- 「Business Mate」標準装備
- 20MバイトHD搭載
- フリーストップサイズ
- 小さいボディに高性能

周辺機器

3.5インチフロッピーディスクドライブ
UE-1F04 定価 ¥49,800
一体型外部バッテリー
UE-1X07 定価 ¥26,000

表計算ソフト

Microsoft EXCEL Ver.2.1 定価 ¥98,000

ワープロソフト

一太郎 AX 定価 ¥68,000
書籍 AX(UE-6Z10) 定価 ¥49,800

AX286N-H2

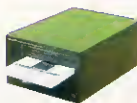
定価 ¥398,000

★発売記念特別価格にて提供中!! 詳しくはお電話で!



X68000用ハードディスク

シャープ
光磁気ディスクユニット
CZ-6M01 予約受付中!
SCSIポート
CZ-6B51 予約受付中!



アイテック

IT X640 定価 ¥158,000

特価 ¥89,800

IT X680 定価 ¥198,000

特価 ¥118,000



PROII CZ653C 定価 ¥285,000
CZ663C 定価 ¥395,000
●次世代のインテリジェンス、SX-WINDOW搭載 ●知的ニュースタンドフォーム ●BIOSの改良によりハイスピード処理を実現 ●2Mバイトの大容量メモリを標準装備 ●拡張I/Oポート4スロット標準装備

EXPERTII CZ603C 定価 ¥338,000
CZ613C 定価 ¥448,000
●次世代のインテリジェンス、SX-WINDOW搭載 ●豪華のフォーム、マンハッタンシェイプ ●BIOSの改良によりハイスピード処理を実現 ●3Mバイトの大容量メモリを標準装備

SUPER HD CZ623C 定価 ¥498,000
●次世代のインテリジェンス、SX-WINDOW搭載 ●「チタン」カラーのクリエイティブラック ●80MBハードディスク搭載 ●世界標準 SCSI インターフェイス標準装備 ●BIOSの改良によりハイスピード処理を実現 ●3Mバイトの大容量メモリを標準装備

Software tools

GRAPHIC TOOLS

マジックパレット 特価 ¥16,830
Z's STAFF PRO-68K 特価 ¥49,300
サイクロンExpress α88 特価 ¥83,300
デジタルクラフト 特価 ¥33,800

電子手帳ソフト

CYBERNOTE PRO-68K 特価 ¥16,830
Stationery PRO-68K 特価 ¥12,580
※通信ケーブル CE-300L 特価 ¥2,520

通信モデム PV-A24MNP5 ツクモ特価 ¥29,800
(消費税別途 ¥894)

ソフト た〜みのる2 ツクモ特価 ¥15,000
(消費税別途 ¥450)

X68000用メモリーボード

一流メーカー

1MB増設用RAMボード ツクモ特価 ¥19,800
(ACE&PROシリーズ内蔵用IMB)

2MB増設用RAMボード 定価 ¥50,000 ツクモ特価 ¥42,500

4MB増設用RAMボード 定価 ¥88,000 ツクモ特価 ¥74,500

※2MBと4MBは全てシリーズ対応拡張スロット用

ツクモグローバルカード

国内・外で活躍!

使って便利、持って安心! ツクモグローバルカードはジャックス・VISA、セントラル・マスターとの提携カードです。ツクモ各店でのお買物がらくらくできる上に、国内はもとより海外でのショッピングもOK、しかも18歳以上なら学生でもOK!



お申し込みは(03)251-9898又は各店で

秋葉原各店

7号店

5号店

通信販売部

中央通り

秋葉原駅

AM10:15~PM7:00 (毎週木曜日と8/15)

★表示価格には消費税は含まれておりません。

ツクモは「スーパーX PRO SHOP」です。

PRO STAFF

九十九電機株
〒101-91 東京都千代田区神田郵便局私書箱135号

★商品のご注文は在庫確認の上お願いします。



7号店 船井



N 店 福地

ツクモ7号店 ☎03-253-4199 (担当/荒井)

便利で安心な通信販売
通信販売部 ☎03-251-9911

- ニューセンター店 ☎03-251-0987 (担当/福地)
- ツクモ5号店 ☎03-251-0531 (担当/川名)
- 名古屋1号店 ☎052-263-1655 (担当/吉高)
- 名古屋2号店 ☎052-251-3399 (担当/横山)
- ツクモ札幌 ☎011-241-2299 (担当/村井)

カード払い	全国代金引き換え配達	クレジット払い	現金書留払い	銀行振込払い	各種リース払い
通信販売での御利用カード、ツクモグローバルカード、VIPカード、セントラル、ジャックスを御本人様より電話で通信販売部へお申し込み下さい。	お申し込みは ☎03-251-9911へ お電話1本! 配達日の指定もできます。	月々 ¥3,000以上の均等払いも 頭金なし、夏・冬ボーナス2回 払いも受付中!	〒101-91 東京都千代田区神田 郵便局私書箱135号 九十九電機株通信販売部 oh/X係	事前に ☎で御届け先をご連絡下さい。 富士銀行 神田支店(普)N894047 九十九電機株	くわしくは各店にお問い合わせ下さい。ケースに合わせてご相談にのらせて頂きます。

冬のボーナス一括払受付中! くわしくはお問い合わせ下さい。

表紙ぎゃらりい

1982年5月18日の創刊以来、本誌は誌名を変えても変わらぬ心で誌面作りを続けてきました。応援して下さった読者の皆さん本当にありがとうございます。お

かげさまでOh! Xは通巻100号を数えることになりました。ここにその表紙のすべてをご紹介します。これからも本誌をよろしくお願いいたします。

①創刊号



②7月号



③8月号



④9月号



MZ専門誌としてデビューしたOh!MZ。創刊号は104ページで620円。あまりに高いとの声に次号から480円に値下げしたが……。ちなみに表紙はマジックパス、オウスターなるヒロインが活躍した。まだXIが誕生する前の時代である。

⑤10月号



⑥11月号



⑦12月号



⑧1月号



パソコンテレビXIの登場で誌面に緊張感が。だが、誌名までが変わってしまう事態を予想した人はどれだけいたであろうか。時はMZ-700の全盛期。一時は読者の4割を超えることもあり、本誌は飛躍的な部数アップを記録した。

⑨2月号



⑩3月号



⑪4月号



⑫5月号



⑬6月号



⑭7月号



⑮8月号



⑯9月号



⑰10月号



⑱11月号



12月号



4月号からあのシド・ミードが表紙を飾る。増ページと共に内容も充実し、ほぼ現在のスタイルを確立。そして11月号には新製品X1 turboの歴史に残る大特集が。MZユーザーの目がこれ以来反感から羨望へと変化したという。

1月号



2月号



3月号



4月号



5月号



6月号



7月号



8月号



9月号



10月号



11月号



12月号



感動のX1 turbo特集

1月号



2月号



3月号



4月号



全機種共通システムS-OSがスタート。また、満開一号を発表(?)した祝一平氏が「試験に出るX1」を連載。時代はその筋へと流れていく。OH!MZがユーザーと共にあるべきパーソナルコンピューティングを追求したのはこのころだ。

5月号



6月号



7月号



8月号



9月号



42 10月号



43 11月号



44 12月号



35 ADVANCED MZ-700



本誌唯一の別冊。発売
が遅れてMZ-700のユー
ザーをやきもきさせた。

45 1月号



46 2月号



47 3月号



48 4月号



読者参加を強く呼び掛ける
特別企画「GAME OF THE
YEAR」"言わせてくれなくちゃ
だワ"を開催。このま
までは世のパソコンがすべて
実務マシン一色になるとい
う不安のなか、ついに夢の
マシンX68000が衝撃のデビ
ューを遂げたのだった。

49 5月号



50 6月号



51 7月号



52 8月号



53 9月号



54 10月号



55 11月号



56 12月号



57 1月号



MZ-286iを機にMZグループ
がビジネスコンピュータへ
の路線転換、パーソナルユ
ースはXfamilyに絞られる。
そのため本誌は12月号でOh!
MZ-Oh!Xと改題した。なお、
1月号から翌年3月号まで
の表紙イラストは永沢しげ
る氏が担当。

X68000が初登場!

58 2月号



59 3月号



60 4月号



61 5月号



62 6月号



63 7月号



64 8月号



65 9月号



66 10月号



67 11月号



68 12月号



X68000ユーザーが増えるなか、本誌では創刊6周年企画として8TRON計画を発表、昭和70年代を目指した究極の8ビットパソコンの姿を考えた。結局70年は来なかったが……。また、4月号からは画家の松葉口忠夫氏に表紙絵を依頼。

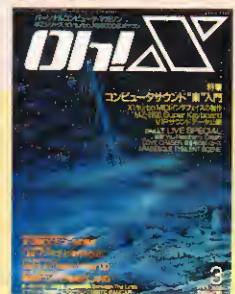
69 1月号



70 2月号



71 3月号



72 6月号



きわどい内容が満載

73 4月号



74 5月号



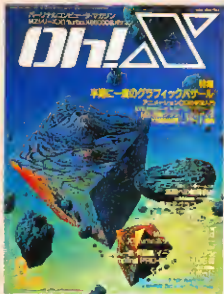
75 7月号



76 8月号



77 9月号



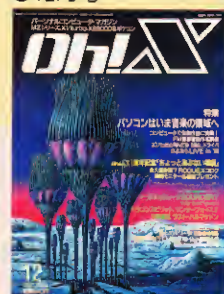
78 10月号



79 11月号



80 12月号



81 1月号



82 2月号



83 3月号



名実共にパーソナルマシンの一大勢力に成長したX68000。読者の割合も半数に達し、誌面もX68000を中心にゲーム、クラシック、サウンド関係の華々しい記事が目立つようになる。4月号からの表紙はもとのりゆき氏にお願いした。

84 4月号



ぶっとんだ
ゲーム特集
が衝撃的!

85 5月号



86 6月号



87 7月号



88 8月号



89 9月号



90 10月号



91 11月号



92 12月号



93 1月号



94 2月号



95 3月号



96 4月号



4月号から表紙デザインを一新。須藤牧人、塚田哲也両氏のCGが交互に本誌を飾るようになった。さて、'90年代のOh!Xは、などと能書きを書いている暇はない。我々はリアルタイムに動いている。Oh!Xはどこへ行くのか?

98 6月号



95 5月号



97 7月号



100 8月号



1月号付録のX68000ゲームソフトウェアカタログ



6月号付録の創刊8周年記念PRO-68K

お祝いの言葉

へーえ、100号? そうか、まだ100冊しか出てなかったのか、もっといっていると思ってた。まあ、100冊たって、数字なんてどーでもいいことさ。さるお方の結婚式ももうすんだし。過去も未来も似たようなもの。大事なのはその100冊に散らばる過去の名作たちだ。逆立ちしてもOh!X(Oh!MZ)でしか読めない、機種の壁を越えた名作・奇作・珍作の嵐。これが財産である。Oh!X傑作集を出したいくらいだ。

いま、その個性も矢面に立たされている。浸透は常に拡散を伴うからだ。いくつものベクトルを内包した新しいスタイルも必要とされるだろう。しかし、知識より知恵、実用より心、完成されたプログラムよりマシンポテンシャルの開拓精神の基本は変わらない。X68000はまだまだ深いポテンシャルを秘めている。のんびりしている暇はない。そして粋なパソコン誌として、多様化する読者と共に、Oh!Xは100万部を目指すのである。

からころも きつつなれにし つましあれば
はるばるきぬる たびをしぞおもふ
てなもんだ。めでたいな。

(荻窪圭)

SOFTWARE INFORMATION

今月は夏休みに向けてか、ひきびきに大量の新作の情報が入ってきました。てなわけで、今回は4ページでお届けすることになります。しかし、毎月コンスタントにこのくらい発表されればありがたいのに……。



ギャラガ'88

2, 3年前だけどゲーセンで流行ったこのゲーム、いよいよX68000にも登場だ。ゲーセン版の移植のみならず、X68000オリジナルの面もあるぞ。



話題のソフトウェア

いや～、先月は梅雨だなんて書いてしまったもんだから、皆さんからのお叱りのハガキの多かったこと。まあ6月18日を予想して書いているんだから、そーゆーこともたまにはあるわな。許せ許せ、ハハハ。というわけで、今月こそ梅雨です。じつにうつうつですね～（え？ フォローになってないって？ でも、梅雨明けって7月22日って気象庁が言ってるからいいじゃない）。そういや、もうじき夏休みですねえ。クーラーの効いた涼しい部屋でアイスティでも飲みながら、ゆったりとゲームに浸る。う～ん、極楽極楽（とか言っってすっかり違う方向へ話を持っていくヤツ）。悪いことは全部忘れて、夏休みの前半は遊びまくりましょ。宿題そのほかで青くなるのは、来月号が出てからでも十分なんだから……（ホントか、おい）。

さて、夏休みを目前に控えて、ゲームのほうもバタバタと活気を増してきました。

なんともうれしいぢやあ～りませんか。うれしさ爆発、ページも倍。これを書く側としては、ほんとに喜んでいいやら悲しんでいいやら……。ま、そんなこと言っっててもしょうがないんで、順を追って紹介していくことにしましょう。

まずはこのギャラガ'88。電波新聞社よりすでに発売されているので、もうクリアしちゃった人もいるんじゃないかな。このゲーム、3年ほど前にゲーセンで流行ったナムコのシューティングなんだけど、たった3年前なのに第一印象で“懐かしい！”と感じてしまいました。もっとも私の場合はこのゲームの元祖、ギャラクシアン（死語だよなあ）を中学生の分際ながら（あん、年がバレる）ゲーセンで遊んでたから、そのとき印象が強いからかもね。で、肝心の出来ですが、これがなかなか。プーッとふくれるハエさんや、かわいいボーナスステージのギャラクティックダンシングもゲーセン版同様のいい味出してます。さすがに先に移植されていたPCエンジンよりは、グラフィックもきれいですし。これはゲーム自体は、そう難易度の高いシューティン

がんばで、くだだ!

1	ボビュラス	(前回順位)	1
2	グラナダ		4 ↑
3	ワンダラーズ・フロム・イース		3
4	ダンジョンマスター		2 ↓
5	天下統一		—初
6	スーパーハンガオン		—↑
7	ジェノサイド		10 ↑
8	三国志II		5 ↓
9	サーク		6 ↓
10	ソーサリアン		7 ↓

疲れたー。いつもはサンプリング抽出をして

るのに、今月は28日までのハガキを全部カウントすることになってしまいました。手伝ってくれたみんな、ありがとね。

さて、100号記念(かどうか知らんが)の完全集計版TOP10。ランクアップ・ダウンもつけてみたけどどうでしょう。

おやおや。そろそろみんな解き終わったと思ったらダンジョンマスターは4位まで落ちてしまったぞ。みんな結構ドライだな。代わって2位の座を手に入れたのは、グラナダ。これはウルフ・チーム最高順位！ イースファンのみならず、もう少しだったのに、残念でしたね。

そして、5位初登場天下統一。このゲームの



パズニック

ゲームでもなかったの、ゲーセン版のほかに、X68000用にオリジナルステージも用意されています。こちらもぜひプレイしてみたいですね。

さて発売中といえば、ブロードバンドジャンプのパズニック。こちらもゲーセン版（タイトー）からの移植です。ゲーセンではじっくり考えているヒマがなかったので、かなりお金を注ぎ込んだ人もいたことでしょう。同じマークのブロックを隣接させて消していくパズルゲームなのですが、ブロックは重力の関係で上にあげられないし、でもってタイミングが命の面もたくさんあるし、一筋縄ではいかず悩むわけなんです、これが。家でじっくり楽しめるようになれば、クリアも夢じゃなくなるかな。でもムリかな、私バカだから。

でもって、同じパズルゲームであるコナミのクオースももう発売されていますね。こちらもゲーセン版からの移植もの。ゲームボーイなどでも発売されているし、けっこうやり込んでいる人もゴロゴロいるのでは？ このゲームはシューティングの要素も含まれているので、ちょっとだけ反射神経が必要かもしれないけど……。

ん？ こうやって書いていくと、なんかゲーセン版からの移植ものばかりだわね。ま、いっか。ついでだから、このまま続けて移植ものを一気に書いてっちゃおうと。

じゃ、次、サイバリオン。このゲームはドラゴンを操って、矢印の指し示す方向へ

評判は……あれ、ハガキはAFTER REVIEWに行っちゃったの？ じゃあすいません、そっちを見てちょうだい。

その下に謎のカムバック、スーパーハンクオン。確かに長く遊べるが、なぜ今になって……。さらに7位ジェノサイドのランクアップも謎だ。もうすぐラグーンも発売されるというのに……。そういや、みんなCDはもう買ったかな。

あやや、三国志IIもソーサリアン（まだいる！）もランクダウンか。先月威張ったのが反感を買ったかな？ こりゃおいらは静かにしてたほうが良さそう。……（それじゃ、また来月）。

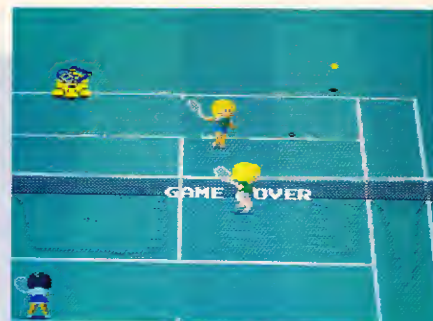
（浦）



サイバリオン

進んでいくタイトーのアクションゲームなんだけど、ゲーセン版はスティックじゃなくて、トラックボールでってところがミソだったよね。今回はジョイスティックでもできるようにになっているけど、通ならやっぱりトラックボールで遊んでほしいな。ジョイスティックに慣れているからこそ、トラックボールで遊ぶっていう感覚は新しくっていいかもしれないし。8月中旬にシャープから発売される予定。いま頑張ってるSPSさんが移植しているの、楽しみにしてて。

でもって、同じくSPSさんの移植によるナムコのワールドコートの登場です。このゲームってば、地味なスポーツゲームと思いきや、結構ハマりやすいゲームだったりするわけ。その当時は友達同士で遊んでいる高校生や予備校生をよく見掛けました。そうこうする間に、PCエンジンにも移植されちゃったりなんかしました。さすがに今回はクエストモードはないみたいだけどね。スマッシュやサーブがうまく決まるようになると、もうまさにテニスの選手になった気分です。そういや、わざと女の子の選手を転ばせてパンチラを楽しん



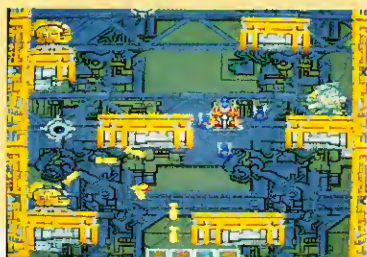
ワールドコート

でいたふとどきものもいたっけかなー。まあ、それはおいといて、このゲームは7月20日に発売される予定ですのでお楽しみに。

さてお次は、じゃーん、イメージファイトなんですね。このゲームはかなりムズかったんで、わりとマニア受けしていたシューティングです。アイレムさんのゲームはあのR-TYPE以来だから、このイメージファイトの登場を待ち望んでいたユーザーも結構いるはず。その夢がやっと実現しました。このゲーム、ボッドと呼ばれるアイテムを、いかにうまく使いこなすかがカギとも言えるでしょう。これをうまく扱えないと、かなり苦しい。はじめてやると全9面クリアどころか、5ステージクリア後にある補習ステージにたどりつくのにもとこずったりするんですよ、これが。で、移植の出来はというと、画面写真を見ての通り。なかなかよさそうでしょ？ コンテニューもあるらしいから、ゲーセン版では見ることができなかったエンディングも見られるかもしれないぞ。年内発売の予定だから、詳しい情報はもうちょっとだけ待っていてね。

イメージファイト

これまたゲーセンで人気だった超ムズいシューティングゲーム。なかにはゲーセンで血を流した人もいたとかいえないか……。





ラグーン

ジェノサイドで人気のソフトハウス、ズームの期待の第2作。今度はアクションRPGだぞ。2頭身のキャラクターがなんとも可愛い。期待度大のゲームだ。



実戦ビリヤード

まあ、ゲーセンからの移植情報はこんなもんな。もうちょっとすると、またいくつか出てくるみたいだけど、それはそれでまたあとのお楽しみということで、ね。

じゃあ、今度はゲーセンものではないやつをガシガシ紹介していくことにしましょうか。

まずは、皆さんお待ちかねのズームのラグーンからいきましょう。ジェノサイドで一躍人気者となったズーム。その第2弾といえば、アクションゲームファンでなくとも気になるところ。開発状況はわりとよいようで、発売に向けて着々と進行している様子です。今回は、最終段階に入ったともいえる現時点での画面写真をお届けしましょう。ジェノサイドであれだけ頑張ってくれたズームが、アクションRPGという新境地でどういった展開を見せてくれるか、楽しみにしたいですね。

さて、バトルチェスでX68000に参入したパック・イン・ビデオからは、実戦ビリヤードが発売中。このゲームは、その名の通りビリヤードゲームで、ナインボールやローテーション、はたまた4つ玉(知ってるかな?)までプレイできちゃいます。プールバーなるものが乱立したビリヤードズームはもう過ぎてしまいましたが、本来

ビリヤードというものはじっくり玉筋を読んで楽しむものだし、家でゆっくりビール片手にパソコンに向かって楽しむのもいいんじゃないでしょうか。

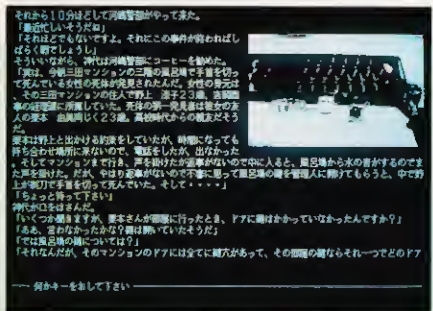
じっくり楽しむといえばやっぱりMisty4でしょうか。一連のMistyシリーズの第4弾です。前作からしばらく間が空きましたが、やっぱりデータウエストさん、頑張ってくれました。今回もユーザーからのシナリオ5つを中心に構成されてます。暑い夏に、ちょっとサスペンスタッチの推理ゲームを静かに楽しむ、なんて大人っぽくない。ところでデータウエストといえば、ピンとくるのが第4のユニットシリーズ。ブロンウィンファンの皆さん、ご安心を。シリーズ第5弾D-Againも着々と進行している様子。今月はまだ画面写真をお届けできないけど、もうちょっとしたら詳しいことをお伝えできそう。待っててね。

でもってT&Eからはルーンワース〜黒衣の貴公子〜が発売、ドラマチックな展開で進んでいくアクションRPGです。なぜドラマチックかというと、このゲームはプレイヤーの行動によって、たどるストーリーが変わっていくからなんです。いわば、あなたがストーリーを作り上げていくゲームなのです。うん、これは奥が深いぞ。

またT&Eでは次回作幻獣鬼を開発中。これはサンプル版をプレイしたところによると、敵の攻撃が、というか敵の放つ弾が雨アラレのごとく飛んでくるので、なかなかタイヘン。やりがいがあるようです。そのほか、あのゴルフゲーム遙かなるオーガスタも出す予定だそうだし、今後のT&Eの動向には目が離せない!?

さてさて、数々のラインアップを控えているザイン・ソフトでは、ただいまREINFORCERとバルーサの復讐をしゃかりきになって開発中のよう。REINFORCERのほうは、トップビュータイプの8方向スクロールという、サイバーパンクアクションゲームだそう。こちらは先月号でも紹介しましたが、さらに開発が進んだものが手に入ったので紹介しちゃいましょう。発売は9月上旬の予定。一方のバルーサの復讐のほうは、剣と魔法で攻撃するファンタジーアクションゲーム。サイドビュータイプで、8方向多重スクロールするというシロモノ。こちらは7月発売を目指して、目下頑張っている中とのこと。お楽しみに。

あつ、とついうっかり忘れそうになっちゃった、いまや読者の人気ナンバー1に輝いたポピュラス。そのポピュラスの追加シナリオが発売になったことは、きっともう皆さん周知の事実でしょう。今号のREVIEWでも紹介していますしね。まあ、それはおいといて、なんとそのポピュラスを発売したイマジニアから、シムシティが移植、発売されることが正式に決定しました。



Misty4



幻獣鬼



REINFORCER



闇の血族

サコムのノベルウェアシリーズ。推理探偵もので主役はうら若き乙女。リアルな感じのグラフィックが雰囲気を出しているよね。



わーい、パチパチパチ。このシムシティー、都市開発を題材にしたリアルタイムシミュレーションで、14個のアイコンを駆使して町を発展させることが目的。鉄道を敷いたり工場を建てたりとなんとなく忙しい。まあ、詳しいことはまた来月にでも紹介させていただきますのであしからず。へへっ、出し惜しみしちゃってごめんね。また、イマジニアではポピュラスの原作者であるピーター・モリニュー氏の来日を記念して、ポピュラス大会を企画しています。我こそは、と思うポピュラスマニアの方、んあ？と思ったらプロミストランドのREVIEWの左下を見て、応募してください。よろしくね。

さて、移植といえばスタークラフトのトンネルズ&トロールズ。こちらもすでに発売になりましたね。もともとテーブルトークRPGということで、そのあたりが好きな方には熱狂的な支持を受けているゲームですが、ようやくX68000にも登場。ほっとした方もいることでしょう。このゲームは、背景となる舞台設定がしっかりしているので、はじめてRPGをやる人でも親しみやすいかな。それにオマケとしてオリジナルオーナーズカードや、ドラゴン大陸のポスターなど、RPG必携3点セット(!)なるものが付いてくるなど、ニクい心配りがうれしいじゃありませんか。毎日コツコツとたゆまぬ努力をしても苦にならない方は、ぜひプレイしてみても?

あちらものの移植じゃあないけれど、こちらも移植もの。PC-9801からの移植だけれど、システムソフトから遊撃王IIがでるそうです。PC-9801版ではサイバースティックが使えるってんでびっくりなのですが、当然のことながらこのX68000版でもサイバースティックが使えます。フライトシミュレータゲームなので、サイバースティックを使えば、パイロット気分が楽しめそう。画面写真もお届けできなかったし、発売はまだ未定だけれど、出来はか

なりよさそうですよ。期待度大です。

さてと、そいじゃシステムサコムだ。ジェミニウイングの開発も佳境に入ったカンジなのだけれど、その一方であのノベルウェアシリーズである闇の血族の開発も、しっかり進行している様子。今回は女の子が主役のアドベンチャーとあってか、サコムとしても主人公のグラフィックにはリキを入れているよう。届いたばかりのグラフィックの数々を紹介しましょう。この闇の血族は、7月か8月には発売されるそうなので、ノベルウェアファンは見逃せませんね。

さて、最後を飾るのはM.N.M.Softwareです。今回紹介するのはThriceとPipyan。ThriceはColumnsタイプのパズルゲームで、縦、横、ナナメに同じキャラクタを3つ以上揃えて消していく、得点を競うというもの。なんと300位までネームエントリーができるそう。ふえ〜。でもって、このタイプはざっと画面を見ているだけでは疲れてきちゃうこともあるので、それをな



トンネルズ&トロールズ



ブルーサの復讐

くすためにもある点数をクリアすること、背景がいろいろと変わっていくので、飽きずにプレイできます。8〜9月に発売されるそうなので、Columnsにはまった人はぜひプレイしてみてください。そしてPipyanは、倉庫番のように男の子のキャラクターを操作して、ブロックをうまく組み立てていくといったゲームです。さながら工事現場のようなステージ上で、あたふたと動き回る男の子、失敗するとペコペコと頭をさげたりなんかして、とってもキュート。こちらは、7月中旬にタケルより発売される予定とのこと。ひょっとしたらこの本が出る頃には発売されてるかもね。

てな感じで今月もそろそろネタ切れです。こうやってずらっと書いたあとで見てみると、おや、X1がひとつもない? んなバカな! でもほんと、そうみたい……。なんかとっても悲しいなあ。ああ、X1ユーザーの怒りの声が聞こえてきそう。ではまた来月。



Thrice



Pipyan

THE SOFTOUCH

●大航海時代



ロマンたっぷり 大海原で帆船の冒険

Urakawa Hiroyuki

浦川 博之

「維新の嵐」に続く光栄のREKOEITION GAME第2弾は、中世の帆船の旅をシミュレーションゲームにした「大航海時代」です。貿易、艦隊との対決、数々の使いっぱ(?)を繰り返し、成り上がるのが目的だっ!



X1 turbo用 5"2D版 4枚組 9,800円(税別)
光栄 044(61)6861

ども。親父が船乗りだった浦川です。おかげで家はいろんなオブジェでいっぱい。おさるさんの置物とか巨大な素焼きの風鈴とか、ダチョウの卵とか。こう節操なく並ぶと海のロマンもなにもあったもんじゃない。で、その因縁か、私が光栄の「海のロマンゲーム」、大航海時代のレビューをやることになりました。これは1500年代初頭を舞台とした海洋シミュレーションです。ポルトガル、イスパニア、イスラムによる貿易の主導権争いの真っ只中の頃ですね。プレイヤーは有象無象の商船長の中のひとりとなり、地中海に始まって、アフリカ喜望峰、アラビア、インド、はてはジパングまで航路を開拓し、貿易を行います。

貿易のほかにもうひとつ、貴族の爵位を得るというフィーチャーがあります。主人公の先祖が航海の失敗から爵位を剥奪されたという設定になっていて、お家の復興がプレイヤーの悲願なのです。オーイェー(面白度1)。ライバル国の艦隊をやっつけたり、勅命を遂行したりして国王に認めてもらい、最高爵位まで昇りつめるべくこれまた世界を駆け巡るわけですな。

ややこしそうに聞こえるかもしれませんが、「貿易する“スタークルーザー”」といえわかるかな(もしくは光栄版“WARNING”か?)。

地中海の隣人

私はタ=バスコ=ガマ。ちょいと辛口のいい男。自分ではちょっただけ銀英伝のラインハルトに似てると思っている。親父が遭難して行方不明になったので、家の再興のために大海原に出て一旗上げることにした。といっても、手元にあるのは親父の残した小さい商船だけ。最初はヨーロッパ周辺で経験を養い、財力をつけねばならない。幸い、頼りになる昔の父の部下ロッコがいる。ひとりでもロッコとはこれいかに? ロッコ「ぼっちゃん、禪問答してないでこれからどうするか決めてくださいよ」

じゃあ酒場に行こう。情勢も知らずに積み荷を仕入れちゃ失敗は目に見えてる。

カランコローン。

Yo「あら、いらっしやい。」

ようちゃん、ここのみんなにwellsスーパーマラソンね。

ロッコ「おや、誰か来やすぜ」

男「あんた、リスボンで何か仕入れるんだったら、砂糖を買うといいぜ」

かくして1502年2月、タバスコ一行と砂糖をどっさり載せたラテン船「難破1号」は大西洋へ漕ぎ出した。……誰だ、こんな不吉な名前つけたやつあ。

航海中の画面は下の写真のとおり。1画面が緯度・経度ともに約5度の広さだ。この左側の矢印はなんだろう。

ロッコ「上は針路。真ん中は風力計でさあ。左上の数字が風力で、その下は潮流計」

いまは逆風だな。三角帆だから逆風でもわりと速いんだよな。速い速い……(ゆるゆるゆる), 速い……。おい、遅いぞ。なんだこの遅さは。おまけに夜が明けるたびにディスクはガーガー鳴るし。

ロッコ「この辺りは外洋と違って風がおとなしいですからね。それに海を航行してるのはわしだけじゃねえんすから、処理速度もちったあ遅くなりまさあ」

ぶーぶーいいながら、3日でイスパニアの首都、セビリアに到着。幸い、砂糖は約2倍の値段で売れた。元が安いからあまり大きな儲けにはならないが、楽な航海だったからこんなもんだらう。しかし、どこの港も人の顔が全部一緒だな。旅情ってもんがない。酒場の娘の顔は違うんだけど。ロッコ「なにぶつぶついつてんです。次はどこへ向かいやすか?」

神聖ローマ帝国のビサで美術品が安く買えるようだから行ってみよう。

再びゆるゆると地中海を進む。このゲーム、舵を切るときはメニューを開かなくてはならない。そのたびにディスクアクセスするので、地中海のような入り組んだところを航行するのはなかなか骨が折れる。

十数日の航海を経て、ビサに到着。すいませーん、美術品くださーい。

交易所の親父「美術品は金貨310枚だよ。いくつ買うかね?」

買えるだけ全部。ところで、この美術品って中身はなんなの?

親父「見てみるかい(ごそごそ)。ほら、



航海中の画面はこんな感じ、どんぶらこっと

名物“ピサの斜塔ぶんちん”。いまなら大小の鉄球もつけちゃう」

ガリレオの実験は100年後なんですが……。

ザ・グレートスト・ミッション

半年近く地中海を駆け巡ったおかげでめでたく2隻目の船を購入できた。名前はもちろん“難破2号”。途中酒場で知り合ったオスワルドという男に船長をまかせる。

地中海の主な貿易ルートは次のとおりだ。

- ・リスボン（砂糖）→セビリア
- ・アントワープ（陶磁器）←→ロンドン（羊毛）
- ・ピサ（美術品）←→マジョルカ（穀物）

もともと、港ごとに物価は違うし、ほかの艦隊の取引によっても相場は変動するので、絶対これというパターンはない。それから「イスタンブールの美術品はいいぞお、儲かるぞお」とさんざん吹きこまれたが、ポルトガルとイスラムの仲が悪いので立ち寄っても追い返されてしまった。王様、なんとかしてよ。トホホ。

さて、そんなある日。立ち寄った酒場で見知らぬ男に呼び止められた。

男「よう、あんた。タバスコさんだろ。マ

ジョルカであんたを捜してる奴がいたな」

ロッコ「なんででしょうね、ぼっちゃん？」

デ、デートの申し込みかな？（ずで）

耳を引っばって連れていかれたマジョルカ港では交易所の親父が待っていた。

親父「わざわざどうも。あなたに頼みたいことがあって捜していたんです。実は陶磁器で儲けようと思うんですが、35ほど仕入れてきてもらいたいです。金貨4620枚で仕入れてきてもらえますか？」

わざわざ呼びつけて使いつぱかよー。

ロッコ「そういうことってちゃいけやせん。かなりワリのいい仕事なんすから。それに交易所御用達になれば王様のお目に止まる日も近いですよ」

ぶーぶーいいながら申し出を受け、ヴェネチアで陶磁器を仕入れてくる。さっさと引き渡し、その報酬で飲んでいと……。男「おい、タバスコさんだろ。リスボンで

王様がお呼びだつていう話だぜ」

ロッコ「やりやしたね、ぼっちゃん！ すぐに駆けつけやしょう」

もちろんだ。この家名復興のチャンスを逃がしてたまるか。リスボンに急行だ！

ゆるゆるゆる。リスボンを目指して帆船はのんきに進む。リスボンに着くや否や、一目散に城へ駆けこんだ。

役人「謁見の申し込みか？ しばらく待たれよ。……。陛下がお会いになるそうです」

荘厳な謁見の間に通される。国王が現れた。面を上げる。緊張の一瞬。

ポルトガル国王「おお、そなたがタバスコか。お前を呼んだのはほかでもない。実は羊毛が38必要なのじゃが、そなたに……おいおい、どうしたのじゃ？」

タバスコ、南へ

勅命の使いつぱを完遂した私は子爵の称号を賜わった。あれからイギリス、北欧まで足をのばし、貿易網はイスラムを除いたヨーロッパを網羅している。新たに中型の船を購入して旗艦とし、ポチョムキン号と名をつけた。

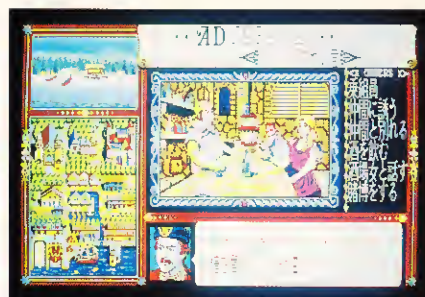
さて、ロッコ、新しい船も手に入つたし、こらでひとつアフリカに行ってみようと思うのだが。あそこじゃ金が手に入るという話じゃないか。

ロッコ「うーん、ちょっと装備が弱い気もしやすいが、いつまでもヨーロッパでもないですしねえ」

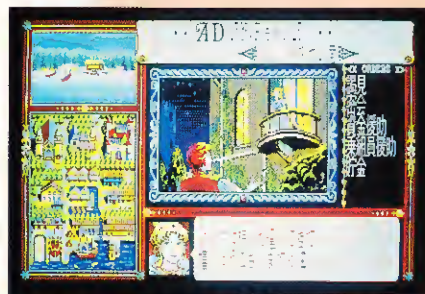
よし、決まりだ。食料と水を満載し、ひたすら南を目指す。セビリアから2,3日ほど行くと海の色も変わり、アフリカに入ることがわかった。ちなみにBGMも変わる。おお、風が強くなってきたぞ。わあ、強い強い。風力8だ。暴風だぞ、こりゃあ。

ロッコ「これが外洋の風でさあ。これに乗って一気に南下しやすぜ」

てててて。信じられないペースで船は進んでいく。うわあ、揺れる揺れる。きばちわるい、げろげろ。ちょっとアフリカは



酒場は大事な情報源、そのほかにもいろいろ……



王女クリスとの密会、たまにはこういうのもね

早すぎたかなあという思いが頭をよぎる。

ロッコ「ぼっちゃん、港が見えやす」

え？ もう着いたの？ まだ1週間そこそこののに。しかし、交易所には金がいっぱい！ 有り金はたいて全部買い込む。はっはっは。帰れば大金持ちだぞ、ロッコ。

てててて。帰日も快調。見事アフリカ金貿易航路が開けたかと思われたが……。

「提督。嵐だ！」、ざざーっ。もりもりと海が盛り上がり、船はひっかきまわされた。「舵がききやせんぜ！」。西を向きながら、船は東へ押し流される。もうムチャクチャ。「難破1号の姿が見えやせん！」

海は一昼夜荒れ狂い、さらに難破2号までが行方不明になった。やはり名前が悪かったか。旗艦ポチョムキンも食料の半分と3分の2近い乗組員を失った。安易に外洋に来るんじゃないかった……。と、放心状態でさまよっていたのも東の間。

「提督。嵐だ！」

この船の末路が私の脳裏をよぎった。

〈ちょっとひとこと〉

貿易が題材のゲームというのは、どうしても単調になりがちです。イベントなどを設定してうまく防いでいますが、操作性の問題が目につきやすい序盤では「ずーっとこんなことが続くのか」と目まいを覚えるということもあります。

地中海を出るようになれば、自分で航路を開く楽しみもあって、自分の好きなように遊ぶことが可能になります。規制が緩く、自分の好きなように遊べるのが身上です。なんだかんだいって結局ハマってしまうのが光栄のすごいと

ころ。

最後にBGMのことでありますが、音楽性がないとはいませんが、「3パートしか使わないBGMを聞かせてCDを売り込むのはちょっと無理があるんでないの」ということは指摘しておきましょう。

冒険心刺激度	10
マニュアル親切度	9
グラフィック	8
操作性	6
BGM	4
熱中度	8



戦闘画面はやっぱりヘックス

●ウルティマV



天下無敵の シリーズ第5弾

Ogikubo Kei

荻窪 圭

その面白さがわかる人にはすごく面白い。そういう一風変わった、しかも奥深い魅力をもつウルティマシリーズの5作目がいよいよ登場。さらなるリアルさと難解な謎であなたの頭を悩ませる？



X68000用 5"2HD版2枚組 9,800円(税別)
ボニーキャニオン ☎03(221)3161

ああ、ダンジョンマスターって、なんて楽なゲームだったんだろう。メモを取る必要はほとんどなかったし、地下6階までは下を目指して進んでいけばよかった。

ウルティマはそんなわけにはいかない。右も左もわからない大陸の真ん中に放り出され、行くも地獄行かぬも地獄、森の木陰でドンジャラホイ、なのである。世界の合言葉は森ってなものだ。

どーして怠慢で出不精で睡魔に魅入られた私がウルティマVなどという超大河、スーパー大河なゲームをすることになったのかというと、ウルティマIV経験者がほかにいなかったからである。経験者っていうだけで終わらせたわけではなく、しかも3年前、友達の部屋のX1turboIIで遊んだものだったりするので、当然育てたキャラクターは持ってこれないし、当時集めた膨大なメモは引越しの際にみんな捨てちゃったしの後悔先に立たず、あとの血祭り村祭り、かんなん汝を玉にするってな状況。人生、蜜のように甘くタバスコのように辛し。

懐かしい風景、旧知の友

イオロ、シャミノ。記憶の底にこびりついた青春の残滓から消え去る寸前のデータベースにこびりついてた懐かしい名前。こんなことまで覚えているなんて。いや、覚えているというより思い出すことができるといったほうが正しい。あくまでも画面にその名が記されたとき、懐かしさを感じるだけだ。役に立たない記憶。

主人公はアバター。アバターというのはAVATAR、アバターとかアヴァターラなどともいう。「化身」とか「権化」という意味である。化身といえばレインボーマン。レインボーマンは月の化身、火の化身など7種類の化身になった。つまり、アバターだったわけである。レインボーマンといえば「インドの山奥で修行」。このインドがポイントでアバターというのはもともとインドの言葉だったのだ。インドにおいてヒンズー教のヴィシュヌ神は人々の前にさまざまな動物や人の姿を借りて現れると考えられ、それを化身(民衆を救おうとして神が姿を変えて現れること、あるいはその姿)、つまりアヴァターラと呼ぶのだ。

で、ウルティマVの主人公はウルティマIVで8つの徳をすべて極め、アバターとなった者なのである。私はなった覚えがないがなったらしいのである。なった覚えがある人(つまりウルティマIVからキャラクター

を移した人)は、それなりのレベルから始められるが、私のようにアバターになった覚えのない人はアバターのくせにレベル2という苦難の始まりとなる。弱い弱い。

舞台はウルティマIVと同じ広大な大陸だ。しかし、前作でとったメモがない。最初からやりなおし。それでも歩いているとだんだんと思い出してくる。ここに村があった、この辺にムーンゲートが出るはずだと。

自由の持つ厳しさ

ウルティマがほかのRPGと異なる点はゲームを進めるためのガイドがまったくないことである。イースを代表とする日本式RPGはスゴロク型であった。ダンジョンタイプのRPGも、その存在自体にダンジョンを深いところへ向かって降りていくという不文律のガイドがあった。しかし、ウルティマは恐ろしい。前向きRPGではなく、はなから、大陸の真ん中で右往左往、どこから手をつけてどこへ向かうのかも自由なのだ。かなりレベルが上がった後半にならなければ行けないような場所でも、然るべき情報と金を出して買えるアイテム(船など)があれば行けてしまうのだ(ちなみに、キーバツファはたまらないぞ)。

つまり、ドラクエやイースやらのスゴロク型RPGが管理された、安全だけど自由のない日本であれば、ウルティマは自由だけど危険ですべて自分の集めた情報を基に自分の判断で動かねばならないアメリカなのだ! ほほほほほう。あなたはどっちが好きですか。自由社会? でも、自由の旗のもとで自由に生きていくためのプレッシャーは相当なものである。

たとえば、ウルティマVではお城のオークの樽に隠されたアイテムを盗むことも、寝ている衛兵を殺すことも簡単だ。本当に簡単だ。しかし、その結果がどうなろうと自分の責任である。特に、ウルティマVは平和で善良な人々ばかりであったIVと違って



やったー、ついに亡霊登場でレベルアップだ

邪悪なブラックソーンの支配下にあるのだ。その中でアバターとしての行動をやり通さねばならない。不当な要求に答えて「持っている金の半分を衛兵に支払う」のも、信念を貫いて「牢獄にぶちこまれる」のも自由だ。

うーん。このゲームは「うんちやうんちやらの自由」を要求するガキの精神に「自由の持つ厳しさ」を叩き込む教育ゲームだったのか。私はもちろん、血反吐を吐きながらも、管理された健全な社会よりアナーキーで自由な社会のほうを選ぶ。日本という平和で安全な社会が好きなのはガイドに沿って大陸を旅するドラクエでもやっていてください。

複雑怪奇な社会

ウルティマVには表の世界と裏の世界がある。表の世界がブラックソーンに支配された圧政の社会であり、裏の世界はロード・ブリティッシュに忠実な人々が集まった、レジスタンスである。レジスタンス、そんなものまであるのだ。アバターである主人公とウルティマIVとともに戦った仲間たち。もちろん、レジスタンスとともに行方不明になったロード・ブリティッシュを捜し、この世界に平和と徳を取り戻すのだ。それが目的だ。それにはアバターはアバターらしく行動せねばならない。ものを盗むな、罪のない人は殺すな、邪悪な者に対しては勇敢であれ。

何が自由だ！ 道徳的であらねばいけないなんて！ 規範だけではないか。しかも目の前にはおいしい餌がぶら下がっているというのに、道徳的であるために自らを律せねばならないのだ。目の前の快楽に弱い荻窪圭はどーしたらいいのだ。

屋と夜

話はがらっと変わる。ウルティマVのうりのひとつに、時間がある。街の住人は朝になると起き、働き、昼になると食事をして、夜になると寝る。だから、買い物しようと思ったら店が開いている時間に行かないと売ってくれない。夜になると門を閉められて入れない街もある。みな働き者で規則正しい生活を送っているのだ。なんと、夜になると会合を開いているレジスタンスの農民もいる。門番の衛兵もちゃんと食事どきや交代時間には入れ替わる。ベッドももちろん住民の数だけある。私は宿屋のない街では他人の家の他人のベッドで休ませてもらう。こんなリアルな街にも「不法侵入罪」はないみたいで、誰も咎めない（これはた



あまり自由を満喫しすぎるとこういう目にあう

んなる皮肉)。

おおむね、圧政者がいても住民は善良である。が、しかし、巡回する邪悪なシャドーロードがいる。シャドーロードがいる都市に入ると憎しみの空気や臆病の気配を感じるの、そんなときはその都市はやりすごすのがいい。シャドーロードがいる都市の衛兵は私を見かけると有無をいわず逮捕し、商人は金をちょろまかし、住人は会話しがてら何かを盗む。シャドーロードに捕まったら大変で、まず勝てない。しかし、悪いのは衛兵や住人ではないので、怒ってはいけない。

最後に、ウルティマVで遊ぶのに必要なものを書いておこう。

ひとつは根気である。なにせ、スーパー大河であるから。レベルアップも経験値をためるだけではだめで、ロード・ブリティッシュに会わねばならないのはウルティマIVと同じ。ただし、Vではロード・ブリティッシュは行方不明なのだ。そっと教えるとキャンプ中に亡霊が現れてレベルを上げてくれることがあるのだ。うーん、根気の野外キャンプである。

続いて異種世界、異種文化を楽しむ心である。優れたファンタジーはリアルな異文化を持った世界が描かれている。読者はその異文化を楽しむのである。劣ったファン

総評だべさ

良くも悪くも、伝統と格式に守られた底の深さと指10本を駆使する操作性はウルティマである。誰の文句も許さない強さだ。ほとんどローリングストーンズのようなものだ。スターウォーズのようなものだ。

世の中にはちょっと聞いた分には耳に優しくノリやすくヒットする歌謡曲や売れ線ロックと、ちょっと聞いただけでは異質で馴染めないけれど聴き込むほどに味の出る名作がある。ウルティマは後者のほうだ。ウルティマワールドに馴染むほど、味が出て、面倒だなんだと文句をいながらつつい大陸をさまよったり会話にうつつを抜かしてしまう。ストーンヘンジ4000年の歴史というか、ケルト人3000年の歴史というか、孔子の儒教2500年の歴史というか、デ



しゃべる馬の「エド」、じゃなくて「スミス」

タジは現実世界をひきずった文化の上に成り立っているため、想像力をあまり要求されず読みやすいが、ファンタジーとしての魅力に欠ける。

さらに、筆記用具である。いつ、どこで役に立つかわからない膨大な情報。あっちへいったりこっちへいったり。メモが必要だ。経験を語ろう。ユーの街から別の都市へかけると、ユーの街の誰それが知っているよといわれた。すぐにでも欲しい情報だったのでユーの街へ戻って尋ねた。すると、君は俺がそれを知っているということを誰から聞いたんだい？ といわれた。そんなことまでメモしてなかったの、また危険な森を抜けて戻り、名前を確認し、またユーの街へ戻った。メモは重要。

それでもって、英和辞典である。なんといっても英語だ。たとえば、立て札や墓碑銘、看板にはルーン文字で書いてあるものがたくさんあるのだ。そして、それを表に従って解読すると英文が現れる。それを訳さねば何が書いてあるかわからないのだ。ほかにも英語がわかったほうがよい場面はある。このルーン文字を訳すのが面倒なことこと。うーん。

では、みなさん、頑張ってください。ウルティマIVをやっていない人でも、終わってない人でも大丈夫です。

イズニーランド35年の歴史というようなそんな重みは重いのである。

5段階評価

ウルティマ度：★★★★★

ロード・ブリティッシュ度：★★★★★

非ドラクエ度：★

非イース度：★★★

道化師殺人事件度：★★

＊

アメリカンジャーニー度：★★★★★

カリブの海賊度：★★★

ジャングルクルーズ度：★★★★

シンデレラ城ミステリーツアー度：★★★★

アリスのティーパーティー度：★

非スペースマウンテン度：★★★★

非スターツアーズ度：★★★★

スプラッシュマウンテン度：まだ見たことナイ

●プロミストランド



我が神が導きたもう 約束の地とは?

Yamada Junji

山田 純二

巷で人気急上昇のポピュラスに、はやばやとシナリオ集が登場。西部劇編やブロックランド編などAmiga版からの移植5つと、イマジニアのオリジナル、江戸時代編の全8編が収録されている。まだ全面クリアしていない人もこれは見逃せないぞ。

5月に発売以降、巷で大好評のポピュラスにさっそく追加シナリオ集が登場。いままでは神と悪魔の対決という設定のみだったから、この朗報にはもろ手を挙げて喜んでしまったわけだ。

この追加シナリオ集には、インディアンと騎兵隊の戦い「西部劇編」、変な宇宙人同士の戦い「シリーランド編」、童心にかえてブロックとたわむれる「ブロックランド編」、ベルサイユのばら（ふっ古い!）を思い出す「フランス革命編」、未来世界での大手コンピュータメーカー同士の争い「ステーションナリーワールド編」、そしてなぜか武士と商人が戦うイマジニアのオリジナル「江戸時代編」と、6つのシナリオが含まれている。で、このバラエティ豊かなそれぞれのシナリオに合わせて、キャラクターデータもちゃんと変更されている。そのうえ、各面の設定条件やコンピュータ側の思考ルーチンにも変更が加えられている。オリジナルに比べると結構難しくなっている。というか、敵が強くなっているといったほうがいいな。

んでもって追加シナリオだから、プロミストランドを遊ぶには、とーぜんポピュラスのディスクが必要になる。これを知らないいまきに宝の持ち腐れと化してしまうので注意すべし。

このプロミストランド、ルールや操作法、使える奇跡などはオリジナルのまま、特に変更はなし。ただ、効果音も同じなのはちよっと残念。プレイしてみればわかるけど、各シナリオごとに特徴があるので、それにあった効果音が欲しくなってしまう。どれをとっても個性がつつんしているとっても楽しいシナリオなので、戦いの音や沼に落ちたときの音がそれぞれ違っていたら、もっとよかったのに……。

この6つのシナリオのなかで、僕が気に入っているのは、江戸時代編での沼地。まるで、肥だめのような零闘気をかもし出して、落ちたらとっても臭そう。敵の民が落ちたときに、僕は今まで以上に、エクスタシーを感じてしまった（ん？ 危ないって?）。それでは、69面までプレイしたなかで、僕の気に入った（はまってしまった）、はたまた印象に残った3つのシナリオを紹介していきましょう。

そちも悪人よのう

ひとつ目は、江戸時代編。このシナリオは、ところどころに桜や松の木があって、なかなか日本情緒しているところが気に入ってしまった。特に面白いのが城の中庭。よ〜く見てみると松の木と玉砂利が敷いてあったりなんかして、細かいところまでやってくれるなあ、イマジニアさん、などとすっかり感心してしまった僕。まだ最初の面だからやりたい放題できるのをいいことに、新しいキャラクターの仕草を堪能しつつ、悪行の限りをつくしてしまった。

まず手始めに、必殺肥だめ攻撃！（うわあ、ディスプレイの向こうから臭ってきそう）もちろん、ただあちこちに沼を仕掛けるわけではなく、周辺に地震を起こして、



江戸時代編。桜も満開できれいだこと

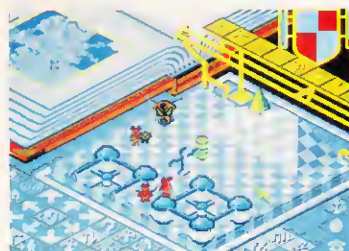
いきなりですが、ポピュラス大会のお知らせ

夏休みにヒマを持て余している諸君、キミの「ポピュラスの腕を試すときがきたぞ!」なんてこんな企画が持ち上がったかという、なんでもポピュラスの原作者であるピーター・モリニュー氏がイマジニアのイキナはからいで8月25日に来日するそう。で、さすがは原作者、対戦ポピュラスにおいては未だ負けたことがないと豪語なさっているらしい。日本のポピュラスフリークともぜひ対戦を、てなわけで、あれよあれよという間にすっかりこの話が決まってしまったのである。

さてさて、この大会には7つのパソコン雑誌チームとイマジニアの計8チームが出場、おのおの読者代表（イマジニアは違いうらしい）をしたがえてこの大会に挑むわけだ。で、トーナメント形式で戦い、その8チームの優勝者がピー

ター氏と晴れて対戦、まさにポピュラスの王者決定戦というわけ。対戦期日は8月18日と26または、27日。まず18日に8チームの優勝者を決定、26または27日にピーター氏と対戦する予定。

そこで、だ、我がOh!Xでもゼツタイの自信と意欲のある読者代表を求めている。我こそはと思ったら、すぐさま官製ハガキを買いに走り、住所、氏名、電話番号、そんでもってこれがいちばん大切なワケだが、CONQUESTモードでの最高面数とそのパスワードを明記のうえ、Oh!X編集部「我こそはポピュラスの王者なり」係まで送ってほしい。応募の締め切りは8月5日（必着）。場合によっては、編集部で腕前を見せていただくのでウソや人から聞いたパスワードは書かないように。それでは、勇気あるポピュラスフリークの応募を待っているぞよ。



X68000用
イマジニア

5"2HD版 4,800円(税別)
☎03(343)8911

相手の民を引きずり出してから、沼を仕掛けるという極悪非道ぶり。そうすると、家から追い出された相手の民が、ボットンボットン、気持ちいいほどよく落ちる。

そうやってしばらく遊んでいると、相手の土地と自分の土地がつながるので、すかさず自分のシンボルであるまねき猫（相手のシンボルは「たぬき」だったりする）を移動して、民を誘導して敵地に突っ込ませる。当然、仕掛けた沼地は、地震と火山で潰しておく。でないと自分の仕掛けた罠に自分の民がはまってしまうという、間抜けなことになってしまうからね。

そのあとは、侍を作って相手の家に放火させてまわってネチネチと相手を攻撃させていったり、洪水を起こしてもう一度いじめ直そうかな、と思ったけど、あまりにも暗いので結局は最終戦争で勝負をつけて終わりにしてしまったのだった。

ぼくらの願いは世界征服だ！

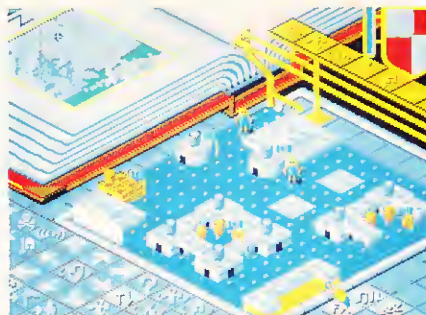
さて、2つ目は子供の頃よく遊んだ覚えのある、ブロックの世界を使ったブロックランド編。マップが見づらいのが難点だが、これといって難しくはなかった。が、しかし53面！これがとにかく面倒だった。最終戦争を起こせないで、勝つためには相手を個別撃破していくしかなく、しかも騎士が作れない。なぜかという、圧倒的にこちらが有利になろうとも、相手を全滅させなければならないので、結局はシンボルを移動させ、リーダーをせっつきながら1つひとつ倒していくという、非常に非能率的な戦法を取らなくてはならないのだ！

攻撃しているときでも、相手はどんどんへんぴな場所に分散してしまうので、鬼ごっこよろしく追いかけて回させられる。そのうえ地面を盛り上げることしかできなくて、土地の整備が難しい。人が増えてくると当然のことながら全体の処理が重くなるため、マウスの誤操作がしょっちゅう起こる。せっかく苦勞して作り上げた平地が、ちょっとしたミスで水の泡になってしまったことが何度あったか。

この面はホント、これら悪条件のためにストレスが溜まってしまった。1時間も2時間もマウスをクリックしていると、肩もこるし目も疲れてくる。まあ、それだけに勝ったときには、すごくほっとしたけど。

哀愁のプログラマ

そして、3つ目のシナリオは46面のステーションナリーワールド編。僕がプロミストランドで初めて負けてしまったのがこの面。



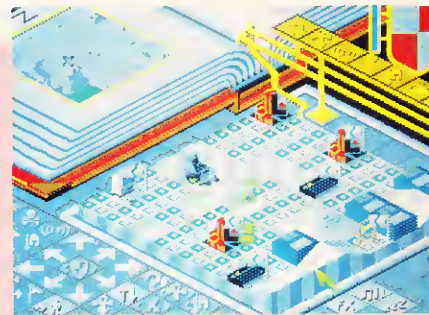
まるでオモチャの国のようなブロックランド編

日頃付き合ひの深いコンピュータ世界での戦いということで、このシナリオは結構はりきって遊ぶぞ！と思いきや……。

今までと同じようにシンボルを移動させながら、相手の土地を目指して進んでいたならば、しばらくして相手の火山攻撃。1発目のときは、わりと余裕たっぷりに、コンピュータも頑張っているなあ、と作られた山を削っていた。が、間髪入れずに2発目の火山攻撃を受けたときにや、マウスを握る手がビクリ。3発目には思わず、マジかよとつぶやき、4、5発目には目が座って、必死に復旧作業をする僕の姿があった。

すでに、連続の火山攻撃で泣きそうになっている状態に、さらに追い打ちをかけるように「ボン」と変な音が。思わず背筋がぞくとして、マップを捜し回ると、いた！ガチャピンナイト（このシナリオのナイトは、まるでボンキッキのガチャピンの頭に足を2本付けたようなやつで、その愛らしい顔とは裏腹に、領土を荒らし回ってくれる）。しばらくするともう1匹、さらにもう1匹と今度は、連続のナイト攻撃！もちろん、火山攻撃も休むことなく続いていて、結局はたび重なる敵の攻撃に耐えられず、負けてしまった。

あまりの悔しさにすぐさま再度チャレンジしたが、結果は同じく負け。しばらく呆然として、設定画面をながめていたら、“WATER IS FATAL”の1行に気づき、3度目の挑戦にして、ようやく勝つことが



こっちはステーションナリーワールド編

できた。わかってしまえばなんのことはない。ナイトは海に沈めてしまえばよかったのだ。ここで初めて、プロミストランドが、難しいと実感した。

500面クリアした人はいるか？

このプロミストランド、それぞれのシナリオは見掛けはおちゃらけたパロディ。が、中身はなかなか手応えあり。それに初めからやり直すのが面倒臭ければ、オリジナルのパスワードが、そのまま使用できるので（サンプル版では）、自分が進んだ面から自由に遊ぶことも可能だ。

欠点としては、キャラクターを変えたことによりマップが見づらくなってしまっていること。ステーションナリーワールドは地面の盛り上がり方が滑らかにつながっているし、ブロックランドでは角張った地面なので、どこが窪みでどこが盛り上がっているか、慣れてくるまで区別が難しいカナ。

それにしても、オリジナルでさえ500面あるのに、さらに追加シナリオが出てしまって、単純に考えたら1000面。発売からしばらくたっているとはいえ、はたして全面クリアした人はいるか。スタッフでは、祝一平氏と西川善司氏の2人が、400面ちょっとのところで争っているようす。ほかには、200面、300面クリアの人がちらほら。しかしまだクリアした人はいないよう。全面クリアしたらどうなるか、気になっているんですけどね。

総評（天国は楽し）

この追加シナリオ集は、それぞれのシナリオに合ったコミカルなキャラクターがわしゃわしゃと動き回り、見ているだけで楽しくなっています。以前、スペースハリアーで、キャラクターデータを書き換えたパロディ版があったのを覚えているでしょうか。あれはただのお笑いの世界でしたが、このプロミストランドはシナリオごとにそれぞれ因縁の対決を再現していてストーリーを感じさせてくれます。

さて難易度ですが、本文中でも述べたと思いますが、“6面から相手はナイトを作れるよう

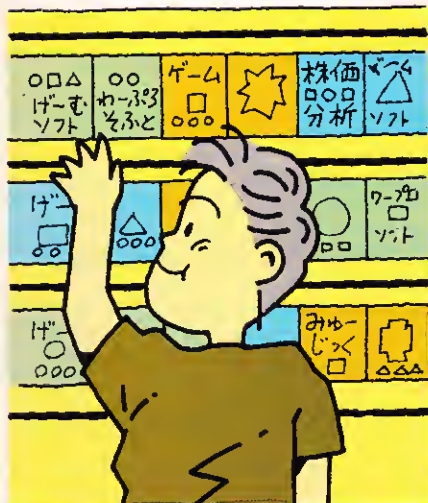
になる”と、一言いえばわかると思います。キャラクター自体は可愛いくて愛敬もあるくせに、やることは手厳しい、まさに可愛さ余って憎さ100倍とはこのことです。先へ進むのは結構タイヘン。体力と時間のある方は、ぜひ挑戦してみてください。

総評（5点満点）

キャラクター	5
肥だめ	5
変身	4
シナリオ	4
難易度	5
やっぱり面白	5

AFTER REVIEW

今月は、天下統一、ダウタウン熱血物語、あーくしゅの3つに加え、8月号の付録ディスクに収録したYet Another Columnを紹介しします。さあ夏休み、思う存分ゲームにひたれるときが来たぞ。この夏やりこんだゲームの感想をどんどん送ってちょ。



天下統一

▶最後の最後まで手が抜けない。最後までライバルといえる勢力が存在する。

岡山県・水口 仁郎(21)

▶私は日本史が好きです。

大阪府・加藤 弓弦(22)

▶現在のX68000のシミュレーションゲームでいちばん楽しめる。

徳島県・中沢 賢一(22)

▶戦国時代のようなすそをみごとにシミュレートしているから。

広島県・平本 裕司(18)

▶コマンドはかんたんだがよくできている！

千葉県・根市 浩(27)

▶反射神経を必要としないし、自分の住んでいる国から統一にかかれる。

滋賀県・小池 清(42)

▶末長く遊べそうだから。

新潟県・保科 康広(20)

▶画面よし、音楽よし、内容よし。

京都府・可見 典明(17)

▶アルシスの移植と聞いただけで……。



熊本県・中村 巧(19)
▶ほかの機種で有名であったが、それがまたいちだんとパワーアップして登場。

鳥取県・安岡 正美(18)

▶戦いが城単位だから戦略的に自由度が高いのが、思ったより面白い。

北海道・近江 弘和(18)

▶戦国シミュレーションファンにはオススメ。

東京都・金子 博政(24)

▶かゆいところに手がとどく
北海道・釜蓋 実(19)

あのアルシスソフトが移植をして、システムソフトが発売した戦国シミュレーションとあって、発売される前から評判だったこのゲーム。フタを開けたらやっぱりこのとおり、の人気でした。フルマウスオペレーションもさることながら、やはりシンプルかつわかりやすい点が、ユーザーの共感を得たのでしょう。統一を目指していく手段も、国対国の争いではなく、1つひとつ城を攻略していくといったやり方なので、ゲームを進めていくうえで、非常にやりやすくてきているといえます。また、余計なものを排除したとはいえ、各々のグラフィックもなかなか見応えがあるものでした。しかも、評価版に比べて製品版はかなりスピードアップしているようです。

こういったシミュレーションものは、まず第一にコンセプトがしっかりしているかどうかにかかっています。シンプルでもいい、面白いものを、というその意気込みがひしひしと感じられ、プレイする側としても、うれしい作品でした。

X68000用 5"2HD版2枚組 9,800円(税別)

システムソフト ☎092(752)3902

発売中のソフト

★ギャラガ'88

電波新聞社の今度の新作は、ナムコの「ギャラガ'88」。「ギャラガ」というゲーム自体は1981年に発表され、未だにゲームセンターなどでちょくちょく見かけるが、このギャラガ'88は、1987年に発表されたそのリメイク版だ。自機を2連結・3連結させて、ギャラガ星人を心ゆくまで吹き飛ばしてちょうだい。X68000版には電波オリジナルのボーナスステージが追加される予定というから楽しみ。

X68000用 5"2HD版2枚組 8,200円
電波新聞社 ☎03(445)6111

新作情報

★遊撃王II

21世紀の近未来の空に展開する、最新鋭戦闘攻撃機「MI-C.A.D.O.II」型の活躍を描くフライトシミュレータ。ミッションブレイクモードのほかにはフライトシミュレートモードが用意され、まず訓練飛行・模擬戦闘でパイロットの腕を磨くことができる。MI-C.A.D.O.IIに慣れたらミッションブ

レイクモードに挑戦。迎撃、偵察、攻撃、護衛の中から任務を選ぶ。弾数や燃料を考慮し、みごと任務を遂行できれば昇格できる。目指せ、最高階級！ サイバースティックにも対応し、フライトシミュレータファンにはたまらない一作といえそう。

X68000版 5"2HD版 予価8,800円
システムソフト ☎092(752)3902

★Thrice

立て続けに新作を発表しているM.N.M. Software。今度はパズルゲームが登場だ。ブロックが上から降ってくるというのはお決まりだが、着地してから回す倒すひっくり返すの大騒ぎ。テトリスでもない、コラムスでもない不思議な感覚のゲームだ。隠れフィーチャー、季節感のあるグラフィック、古代裕三氏のBGM、ビデオ機能に300名までのランキングと盛りだくさんに詰めこんだ、M.N.M.入魂の一作。

X68000用 5"2HD版 価格未定
M.N.M. Software ☎0423(60)3084

★サイバリオン

マニア垂涎のマト、あのタイトーのサイバリオンが家で遊べるようになるぞ。

メカニカルな龍をトラックボール(X68000版ではキーボードなども可)で操り、炎で敵も弾も振

ダウタウン熱血物語

▶お店へ入っているときのくにおやりきがかわいい。戦い方がいろいろあっていい。

長野県・山崎 芳照(15)

▶画面がどう考えてもX68000のものとは思えないが、やってみるとやみつきになる。

茨城県・関根 信男(17)

▶とんでもないマップさえなければ、最高のなあ。

東京都・高見 創(19)

▶ファミコンの移植だからダメかなと思ったが、これが意外と面白いのよ!

高知県・井上 哲郎(24)

▶他人がどう言おうと私は好きや。

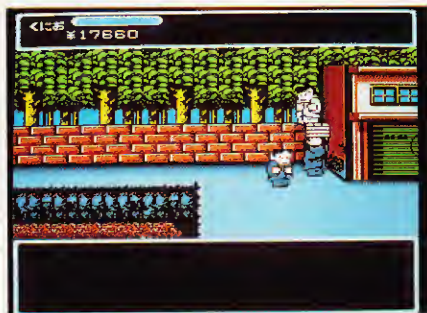
大阪府・渡辺 雅之(29)

たくさんアイテム、殴る、蹴るなど日頃のうっぷんをはらすにはもってこいだったこのゲーム。やはり、そのあたりの単純さがよかったのかもしれない。マップはやや入り組んでいましたが、それがかえてこのゲームを面白くしたともいえるでしょう。

X68000用 5"2HD版2枚組 8,800円(税別)

シャープ

☎03(260)1161



あーくしゅ

▶ピクトのまじめさに対し、じえだのすっとばけた会話がすごくいい。

埼玉県・奥村 光雄(15)

▶じえだが二重人格者だから。おまけに言うとき、マウスカーソルはヤマトとウルトラマンとやじるしもあるぞ。

北海道・谷口 有香(21)

▶短時間で解けるのがいい。

東京都・合屋 琢(21)

このゲームに関してはカワイイ、とか面白いとかいったひととて表せるような感想が多かったですね。いままでのウルフ・チームとはひと味違って、遊びの部分でできあがっているような感覚が、気負いを感じさせずかえってよかったのかもしれませんが、それに、なんといってもキャラクターがみんなかわいい。いずれにしても、ウルフ・チームは、こういったパロディものでも、シリアスものでも作れるという実力を見せつけた作品でした。

X68000用 5"2HD版3枚組 6,800円(税別)

ウルフ・チーム

☎03(5273)4795



Yet Another Column

▶Yet Another Columnにハマっています。テトリスの4段消しのときよりも、Yetの連続して消えていくときの気持ちのよさといったら、もう言葉では表せません。得点は3万点ちょっとなので、努力して4万点突破を目指すぞ!

静岡県・富永 恵隆(19)

▶な、なんなんだYetのあのスピードは(速くなったときのことだよ)。パカヤロウ、キーの反応が追いつかないくらい速く動かすんじゃないか〜!

愛媛県・柳井 敏彦(31)

▶テトリスより熱中してしまった。ヘタに込み入ったゲームよりもシンプルで、なおかつ面白いのはこいつくらいだろうな。感謝であります。

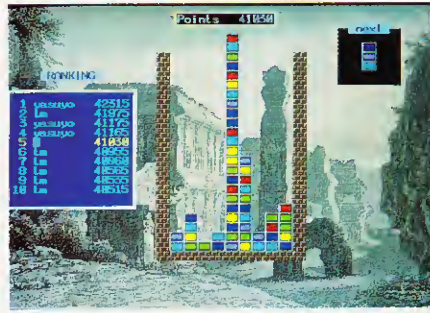
沖縄県・大城 久(18)

▶思いがけない連鎖反応が好き。

福岡県・村上 淳一(18)

まあ100号記念ということで、こんなのも今回は入れてみました。付録でつけたとはいえ、好評を得ているのは編集部としてもうれしい限りです。みんな、遊んでくれますか?

Oh!X1990年6月号付録ディスクに収録



り払ってこれまたメカニカルなボスキャラと対決する。実戦モードでは、プレイごとに独自のシナリオと独自のマップが作られ、ストーリー展開に従ってパワーアップしたり無敵化したりする。おまけにボスキャラのなかには、「ドライアス」のボスキャラも入っているとか。操作感覚に慣れるための練習モードもあるぞ。

移植はSPS、トラックボール対応とくればいやがうえにも期待は高まる。今からトラックボールさばきを鍛えておこう。

X68000用

5"2HD版 価格未定

シャープ

☎03(260)1161

★ラグーン

言わずと知れた「ジェノサイド」のズームが放つファンタジーRPG「ラグーン」がいよいよ発売になるぞ。300年前。7人の魔導士が邪神を呼び出してしまったことがすべての発端となった。3人の命を犠牲にして邪神は封印されたものの、この一件は魔導士の間に決定的な影響をもたらした。邪神の力に魅入られ、その力を手に入れるべく「闇の皇子」を捜す魔導士ゼラー。そしてその闇から世界を守ろうと「ムーンブレードの勇者」を捜す魔導士マティアス。そして彼は少年ナセルとの決定的な邂逅を果たす。彼こそがムーンブレードの勇者なのだ……。

子供が泣きだすほどのデカイキャラと、ゲーマーが腰を抜かす激しいアクションに酔いしれてちょうだい。

X68000用

5"2HD版 8,800円

ズーム

☎011(613)0191

★幻獣鬼

古より、6つの魔界との接点「結界」に囲まれた王国ジタンの人々は魔物と戦う宿命にあった。しかし、ある日無能な魔導士が結界を破り、魔物が王国に攻めこんでしまった。戦士レオン、魔導士リノ、忍者ルイカの3人は、結界を封じる6つのロシュファの魂を奪い返すために旅立つ。MSX専用に開発された「アンデッドライン」が、X68000用にパワーアップしてリリースされる。プレイヤーは3人のキャラを自由に選び、好きなステージから攻略してゆく、キャラによって面のアイテムなどが微妙に変わるなど、数々の趣向を凝らしたT&Eの自信作だ。

X68000用

5"2HD版 価格未定

T & E SOFT

☎052(773)7770

★イメージファイト

つぎつぎとビデオゲームの移植が続いているなか、ついにシューティングゲームの真骨頂、イメージファイトが登場。20XX年、東西陣営の軍事競争のなか、突然西側のムーンベースが大爆発を

起こした。西側未確認の戦闘機によるものと判断した西側は、最新戦闘機OF-1を急きよ用意した。訓練飛行は完全ではないものの、コンピュータシミュレーション試験に合格した者は即、宇宙に飛び立っていった。

最初の5ステージがそのシミュレーション面になっており、平均90%の撃墜率をマークしたものだけが実戦へ進むことができる。落第者は補習ステージ行きだ。ポッドシュートやスピードチェンジ、特殊攻撃パーツを使いこなし、目指すはムーンベース内のマザーコンピュータだ!

X68000用

5"2HD版 価格未定

アイレム販売

☎06(535)4880

★バルーサの復讐

反響を呼んだ「トリトーン・ファイナル」の続編だ。剣と魔法を駆使する8方向多重スクロールのアクションだ。大魔王アレスターに侵略され、国を捨ててウオークの国にやってきたひとりの少女。彼女はウオークにくる途中、突然現れた悪魔により、船は難破し、彼女の兄は呪いをかけられ連れ去られてしまったという。勇者スタイルは、大魔王アレスターの持つムーング石で少女の呪いを解くため旅立った。

X68000版

5"2HD版 価格未定

ザイン・ソフト

☎0794(31)7453

ADVANCED 2D GRAPHICS

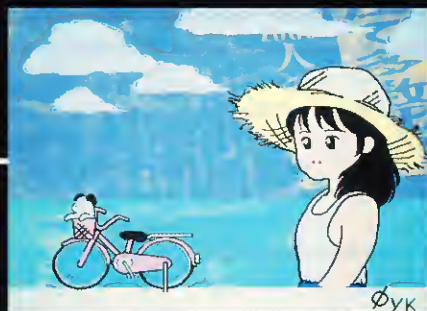
X68000のX-BASICに初めて触ったときのことを思い出す。アナログRGBをサポートしたマシンのグラフィックは……と期待しつつLINEを引いて、表示されるギザギザした線にちょっぴり失望したものだった。

これまでのグラフィック特集ではどちらかといえば3D処理を主体にしていたように思う。これもX68000発売から比較的早期にZ'sSTAFF PRO-68Kが発売されたことが大きい。このツールはそれまでのパソコングラフィックの枠を超えた処理を実現した。そして名実ともにX68000の標準的グラフィックツールとなっている。

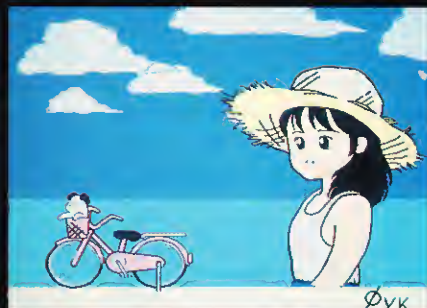
しかし、はや3年。内容はともかく、もはや新しいコンセプトのツールとはいえない。その他のツールもZ'sSTAFFに追いついていない。もっと違ったコンセプトに基づくツールができてもいいのではないか？



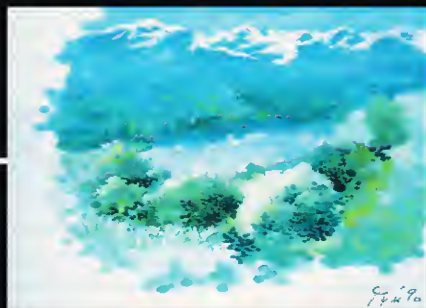
これをスクリーントーンとすると……



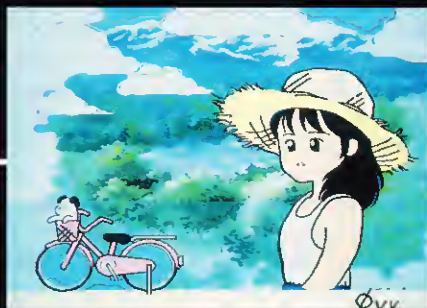
スクリーントーンつきペイント



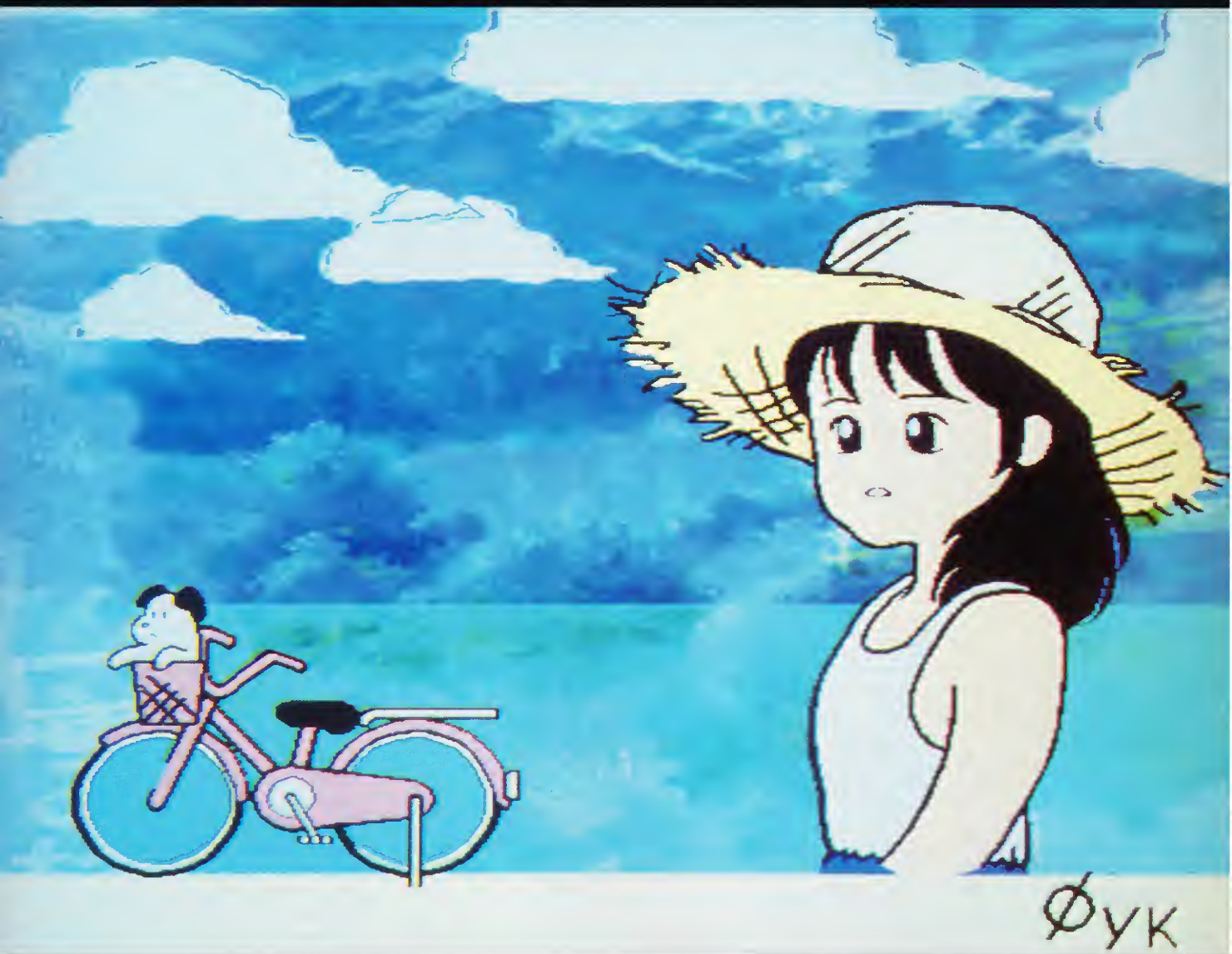
元絵



タイルとして登録し……



タイリングペイント



φyK

アンチエイリアシング対応スクリーン・トーン&タイリングつきペイントルーチンの応用例。タイリングペイントとはいってみればグラフィック・パターンの連続張り付けだ。デジタルRGBでは多色表示のために使われていたが、アナログRGBではあまり使われない。メモリに余裕があれば張り付けるタイルの大きさに制限をつける必要はない。これはヘッダを書き換え、最大512×512ドットの画像をタイル登録できるようにした関数での実行例だ。▼空の部分にスクリーン・トーン（全画面分の新聞紙）とタイル（背景）をペイントした。ビデオなどのクロマキー合成に似ているが、アンチエイリアシング対応なので、マスキング不要で本当に塗りたい部分の隅々までペイントできる。スクリーン・トーンとは合成の比率を決定するもので、パターンさえ用意すればぼんやりとオーバーラップする画像や任意範囲の階調つきマスキングにも使える。

CONTENTS

X68000用グラフィックツール紹介 あなたにあったグラフィックツール	……荻窪 圭	44
ギザギザのないグラフィック関数 アンチエイリアシングとは?	……丹 明彦	50
X-BASICによる画像処理 後処理によるジャギーの除去	……中野修一	66
色数の補間と量子化 グラフィックを変換する	……鈴木康弘	72
4096色→8色変換 Zの画像をX1で	……亀田雅彦	77



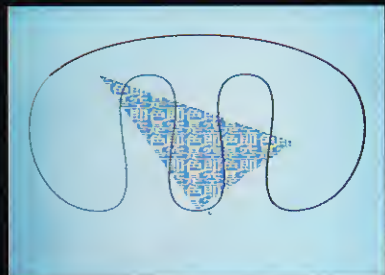
滑らかなラインを見よ



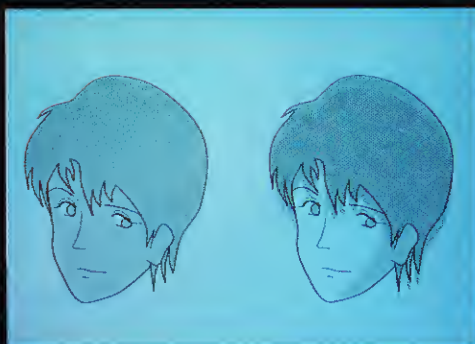
従来の関数による画像



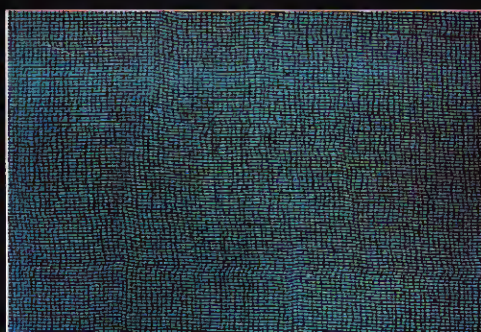
アンチエイリアシングされた画像



半透明のスキャンコンバージョン



スクリーントーンを使う



ガーゼを取り込み反転する



キャンバス地のような表現となる

丹明彦氏によるX-BASICで使えるアンチエイリアシング対応のグラフィック関数の使用例。滑らかなライン（ライン幅調整可能）、ベジェ曲線による滑らかな曲線、そしてタイリングとスクリーントーンに対応したスキャンコンバージョン（閉曲線領域の塗りつぶし）とペイントルーチン。すべてがアンチエイリアシングによる多階調の境界線に対応している。

これらの新しい関数群は単にいままでのBASICにあった関数の発展版としても使えるが、柔軟な思考で使い方を変えれば、さらに新しい可能性が見えてくるはずだ。すでに前ページで行った画面合成。機能が柔軟ならペイントでこういった処理までできてしまう。スキャナを使ってガーゼを取り込んだものをreverse()で反転し、tone_get()でスクリーントーンとして登録。左の写真のキャンバス地のような背景はこうにして作られた。2Dグラフィックもまだまだ面白い可能性を残している。

XROT0によるグラフィックの回転

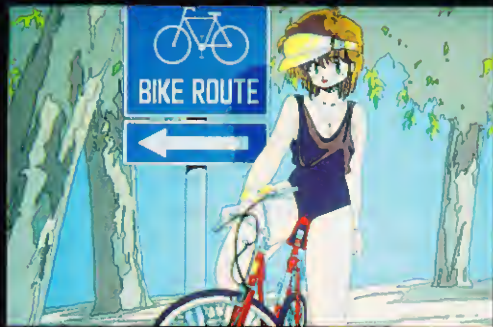
特集の記事ではないが、読者投稿によるグラフィック回転プログラムの実行例。短いプログラムでしかもかなり高速。サンプルプログラムはキー操作により拡大縮小自由自在でぐるぐる回転する。画面下の領域に画面の内容が再帰的に反映されているのも面白い。デモやゲームの特殊効果はもちろん、グラフィックツールの一部として使っても面白い機能だ。



これが元の画像



回転後。下に再帰している部分が見える



元の画像(640×400)



拡大図 タイリングが見える



512ドットで変換



640ドットで変換



タイリングが消えた



これが65536色

65536色の画像をできるだけ原画に忠実に256色モードに変換した例。オーダードディザ法を使ったものと森野雅彦氏のアルゴリズムを応用して多色化したもの。よく見ないとわからないが比較的規則的なパターンになるディザ法と、かなりランダムなパターンになる森野氏のアルゴリズムの違いがパターンになって表れている。階調表現は森野氏のほうが自然に思えるが、もともと白黒2色用なためか、隣接するドットの明暗差が激しく出るのがやや気になる。多色用にアルゴリズムを改善できるのではと思う。誰か挑戦してみしてほしい。



オーダードディザ法



これは森野式

PC-9801のアナログ16色モード用に描かれた画像を×68000の65536色モードに変換した例。もちろんアナログパレットにも対応し、タイリングパターン部分を単色のベタ塗りに変換する。画面サイズを補正して512×512ドットの全画面対応に変換した場合は輪郭線の一部が切られてしまっている。横640ドットにしたものはプロポーションが原画と異なっているが、ドット構成は正確に変換していることがわかる。CRTCをいじれば横768ドットのまますquareに近いピクセルで65536色を出すことも不可能ではない(自由研究)。タイリングが単色に置き換えられるため、65536色化されたもののほうが元ファイルより圧縮がきく場合も多い(PIC, R)。なお、原画は森林林檎氏によるもの。

X68000用グラフィックツール紹介

あなたにあったグラフィックツール

Ogikubo Kei 荻窪 圭

よくこーゆーことをいうやつがいる。
「それで、なにが面白いの？」
ノートに落書きしてるのを見ても絶対そんなことを聞いてきたりはしない。
「で、さあ、それって、役に立つの？」
役に立たなきやいけないときたか。
「プリンタで打ち出したりできるの？」
できねえよ（画面と同じようにはね）。
やっとな、そいつのいうことがわかった。
紙に描いてあったり、ビデオで見られたりしない絵は価値がないというのだ。

グラフィックを描いて遊ぶなんていうのは、コンピュータはなにか役に立つもの、と信じている善良な市民にとって信じられないことらしい。そう考えると、グラフィックツールで遊ぶなんてのは、かなり贅沢な道楽のようだな。道楽万歳。

* * *

目の前に山があるからといって別に登りたいとは思わないが、目の前にあるのが紙とペンだったりするとなにか描きたくなる、楽器だったりするとなにか音を出したくなる。誰も文句はいうまい。目の前にあるのがポピュラスだったりすると締め切りも忘れて沼を作りたくなるというバリエーションもあるぞ。

それでもって、目の前のパソコンにFM音源やAD PCMが乗っていれば鳴らしたくなるし、65536色出るとわかれば色を出したくなる。ポップアップハンドルがあれば持ち歩きたくなるし、ディスクがオートジェクトならゲットイン/ゲットアウトしたくなる、ってなもんだ。それが人情というもので、それが楽しいわけである。

そういったわけで、お絵描きソフト集合である。X68000はワープロよりもグラフィックツールが多いパソコンとして有名だが、グラフィックツールといってもいくつも転がっているわけで、片っ端からあさっていったら私の身がもたない。

で、今月は2次元のお絵描きソフトである。2次元のお絵描きというのはつまり、CRTに投影されているグラフィックVRAM

をべたべたとデータで埋めていくことを目的とした作業のことだ。これがグラフィック画面で遊ぶ基本。

グラフィックモードへの対応

X68000の場合、ご存じのとおり、グラフィックモードをたくさん持っている。

まず1024×1024（表示画面は768×512）の16色。ドットが小さくて、1ドットの縦横比がほぼ1:1である。

続いて、一番メジャーな、512×512の65536色。1ドットを16ビットで表現しようという贅沢さで、512KバイトのグラフィックRAMがたったひとつの画面に収まってしまうという恐ろしいモードである。

さらに、意外とおいしい512×512の256色。1ドットを8ビットで表すわけで、2画面分持てる。さらに、512×512の16色（4画面だ）。

その下に、256×256モードがそれぞれあって、このモードは1画面当たりの情報量が少ないため、高速な処理に向いている。シューティングゲームに多いモードだったりする。

とまあ、こんなにあるわけで（ほかにもいろいろ隠れてたりするけれど）、すべてのモードに対応しているグラフィックツールなんてない、のだ。

順番に見ていくと、まず756×512ドットの16色モード！に該当するグラフィックツールは、なし、である。PDSにもあるという話は聞かない。SX-WINDOWはこのモードのグラフィックをサポートしているので、そのうち出てくるかもしれないが、いまのところ、ない。

これはこれでけっこう綺麗な絵を描けたり、文字を埋め込むには向いているのであってもいいと思うんだが、ないなあ。SX-WINDOWがこのモードだから、もしかしたら、そのうち、マックペイントの玩具みたいなのが出てくるかもしれない。また、PC-9801のグラフィックのちよっと大きい

これまでにX68000用として発売されているグラフィックツールを集めてみました。それぞれの個性や使い勝手について独断と偏見を交えて試用レポートをまとめました。皆さんのツール選びの参考になるでしょうか？ それではサンプルは電腦絵師の福原徹でお送りします。

やつだと思えば、また違ったものが出てくる可能性もある。

さて、512×512ドットの65536色、といえば、Z'sSTAFFと、G68Kである。X68000で一番有名なモードだ。このモードにも欠点があって、それは「メモリをたくさん食う」とか、「ファイルが大きくなる」だ。もちろん自然画を扱おうと思ったら、このモードでないと困るが、自然画は圧縮しづらいのでデータの保存が大変。てなわけでMOディスク万歳。

512×512ドットの256色。実のところ、手描きであれば、このモードで十分な気がする。そこに気がついたのがサン・ミュージカル・サービスであって、マジックパレットという軽快な異色グラフィックツールを出してきた。開発がサン・ミュージカル・サービス、発売がミュージカル・プランという音楽業界コンビのグラフィックツールである。

それから、ウルフ・チームのPRISMもこのモードが中心だ。一応こいつは256×256モードや65536色モードなどもサポートしているが、メインは256色。ゲーム屋さんらしい構成である。

ゲームソフトメーカーというのは、つついグラフィックツールを出したくなるようで（そりゃあ、社内で使うために作ったものがあるはずだし）、ザイン・ソフトからも予定されているようだが、間に合わなかったのとあえずこの4本だ。テラツツォなんてのもあるが、あれはスプライト系なので今回ははずす。

X68000の主なグラフィックツールはこの4つだ。256×256ドットモードのときは、512×512モードの左上4分の1を使えばいいわけだから、問題はない。なかにはちゃんと256×256モードをサポートするツールもある。

画像フォーマット

続いて、とにかくにもグラフィックツ

ールを使ううえで問題となるのが画像データのフォーマットであった。いくらたくさんツールがあっても、それぞれみんな勝手気ままなフォーマットでセーブされたら、たまったもんじゃない！ ってことは、有史以前からいわれていた。クスコーの壁画にも書いてあったほどだ。

X68000の場合、非常に幸運なことに、3つの標準的なフォーマットがある。そのうちの2つはたいいのグラフィックツールがサポートするというラッキーな結果だ。

第1がGL3 (65536色モード時) フォーマットである。ベタフォーマットともいう。X-BASICのIMG_LOAD, IMG_SAVE関数で読み書きできるフォーマットであって、X68000ユーザーなら誰でもこれでセーブされたグラフィックを読むことができる。

ちなみに、256色モードではGM3, 16色モードではGS3, 256×256ドットモードでは3番目の数字が0になる。

この方式の面白いところは、セーブされた画像のモードをファイルの拡張子で区別していること。ファイルには画像データしか入っていないのだ。つまり、どのモードでセーブしたかがデータを見ただけではわからないのだ。私はこういうのはアナーキーで好きだが、無秩序で嫌いだという人もいるかもしれない。この方式をサポートしていないのは、上の4つのうち、Z'sSTAFFとPRISM。G68Kにいたっては、GL3フォーマットを標準フォーマットに採用している。

ちなみに、この形式はもちろんデータ圧縮をしないため、512×512の65536色だと1枚セーブするたびに512Kバイトの磁性面を消費する。ディスク1枚に絵が2枚しか入らないわけだ。

第2が、ZIMファイルである。これは、とにかく権威のZ'sSTAFFである。X68000用で初めてのグラフィックツールで、あまりにメジャーなため、あとから出したソフトはたいいこのファイルを読む機能なり自分のソフトのフォーマットに変換するツールなりをつけることとなった。

圧縮形式と非圧縮形式があり、たいい非圧縮形式をさす。ZIMファイルはX68000に向いているかという、そうではないという意見が大半を占めていて、評判はあまりよくない。

Z'sSTAFF (当たり前だ) のほか、G68K, PRISMがサポートしている。

3番目がPIC形式。PIC.RというPDS (正しくはフリーウェア) の画像データ圧縮・展開ツールの形式だ。圧縮効率が高いのが好まれるところ。でも、PDSなもので、市

販のソフトで対応しているものはなかったりする。自然画を使うのでないならば、とても有効だ。

しかし、どのグラフィックツールも、画面にロードした絵を消さないで起動する方法があるので、ファイルコンバートよりも、こいつを使ったほうが楽だったりする。

では、ひとつずつ簡単にレビューしていこう。

Z'sSTAFF PRO-68K

とにかく、あまりにも有名。PC-9801用のZ'sSTAFF KID-98やらX1turboZについてきたZ'sSTAFF Zからの伝統芸は衰えるきざしなし。伝統の重みはX68000にまで及び、PC-9801なんかと互換性のあるファイルフォーマットを持ち込む (ZIMファイルと呼ばれる) という荒技に出たが、それが唯一の欠点らしい欠点である。

これについては、恐怖の常駐ソフトPIC FILERなるPDS (正しくはフリーウェア) が電腦俱樂部に掲載され、ひとつのマニアックな解決を見せている。これはPIC形式ファイルのロード/セーブをZ'sSTAFF上から行うものだ。

次のバージョンではPICとはいわないが、GL3形式のロード/セーブくらいはサポートがほしい。

メニューは画面一杯開いてまだ余るくらいたくさん開ける。下がPICFILERを使ったところ。本体のみでも自由変形に色変換と機能は尽きない。強い欠点をいえば、マスクのみのセーブができない、2枚の絵を重ねる機能がないというところか



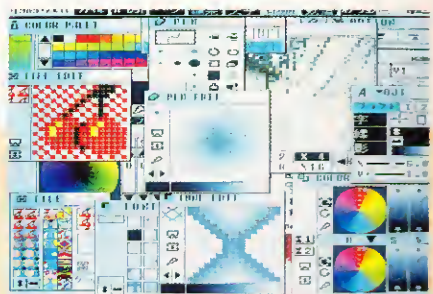
Z'sSTAFFによる作画例

操作の基本は、プルダウン風のメニュー。メニューバー上のメニュー、ファイル、パレット、ペン、編集、文字、印刷、数値、オプションの8つから必要なものをクリックすると、ぽよんとウィンドウが開く。その気になれば、描いたグラフィックが全部隠れるほどウィンドウが開きまくる。

グラデーション、トーン、タイル、にじむ色、自由なペン先、スプライン曲線などお絵描きの機能はやたら豊富。

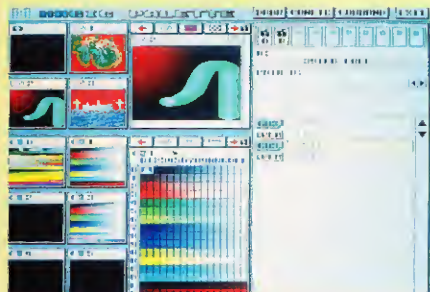
特にそのグラデーションパワーはラインペイント、ボックスフィルや閉曲線ペイントなどいつでもどこでも使え、誰でも描ける富士山とか誰でも描ける円柱などの技を作り上げた。

ペン先やブラシだけでなく、ポップで派手なタイルやトーンなどほとんどのものが編集可能で、特においしいのが濃淡の調節である。ペイントやカラーチェンジも、指定範囲内の色に対して行えるので、各種効





鉛筆画っぽいイメージを目指してみた。セピア調でパレットを統一し、極細ペンを使ってマウスでこりこり……。256色512×512モード固定ながら、なかなか多彩な機能があって使い慣れば相当器用な絵も描けるのではないかなと思わせる。ほとんどの機能がメインウィンドウに収まっているのでわずらわしさが無い。消しゴムもいい。



果が狙える。

編集機能も任意矩形の回転・変形・拡大・縮小、任意曲線内のムーブ・コピー。気になるのは、ムーブしたあとに残る白い跡。背景色が白になっているためだ。

外部入力についてもスキャナからカラーイメージユニットまで対応している。バージョン2からはJIS第1水準のみだが、明朝体とゴシック体のアウトラインフォントもサポートされ、X68000ではどのソフトよりもきれいな漢字が書ける。グラフィックに淡色のグラデーションアウトラインフォント文字を入れると、実に気持ちがいい。

おっと、忘れていたが、一部では致命的ともいわれた「プロテクトモジュール」によるコピープロテクトは、現在発売しているものにはなくなっている。買ったらずい

ていなかったのが驚いた。よいことだ。プロテクトモジュールつきのバージョン2.0を買ってしまった人は残念でした、と。

欠点といえば、プログラムがでかいので、メインメモリが2Mバイトないとアンドゥ機能が使えないことと、アウトラインフォントを使おうと思ったら、ハードディスクがないと大変だということくらいだろう。

512Kバイトの広大なメモリをアンドゥするのは大変だとは思いますが、2MバイトでもRAMディスクをとったり、変なものを常駐させたりしていると駄目である。フリーエリアが1.5Mバイトくらいあれば大丈夫だ。

それから、右ボタンで途中の作業をキャンセルするのだが、「ひとつ前の状態に戻るのではなく、その機能自体がキャンセルされてしまう」のはいただけない。

お絵描きツールのユーザーインターフェイス

いつか祝センセが書いてましたが、ユーザーインターフェイスというものは、たとえ操作しやすくなったとしても、それが古いタイプのものよりも格段にメリットがない場合、人はわざわざ新しいほうに移らないものだそうです。

X68000の場合、最初にZ'sSTAFF PRG-68Kという強力なツールが発表されていたから、後発のソフトは信者獲得には辛いものがあったろうと思われます。僕自身がZ'sSTAFFの虜となっているので、今回の寸評もそこからの視点を中心に書いてしまっているのではないかと少々不安もあつたりします。

が、正直なところ、僕はZ'sSTAFFのようなウィンドシステムは好きではないのです。「下が見えなくて邪魔」なのが主な理由です（これは開発者も感じたらしく、Ver.2ではウィンドウが若干小さく変更されていました）。

グラフィックツールにウィンドシステムはあわない気がします。かといって、ウィンドウ以外に機能を使いやすく配置する方法っていうのが、まだわかっていないんですよね。描画画面を小さくして周りに配置してしまうってのも手でしょうけど、画面を有効に使えなくて悲しいし……。いい方法はないでしょうか。（T.F.）

まあ、ごっちゃんとして、機能と表現力ではまだ他の追随を許さない。Z'sSTAFFの天下はまだ続きそうだ。

マジックパレット

256色モードに目をつけただけでなく、ペインティングソフトとしてのインタフェース構造も新しい。ファイル入出力用メニュー画面。ワープロやエディタみたいなカット&ペースト。アンドゥ用メモリ。メインメモリを2Mバイト積んでいれば、チャイルドプロセスでコマンドシェルを起動できたり、描画画面を3面持てたり。

円のグラデーション（外周から中心へのグラデーション）が派手なおかげで、ほかにもあるユニークな機能は見落とされがちだが、アンドゥ用メモリから任意の形で前のデータを切り出せるとか、カット&コピーバッファも編集できるとか、パレットコード&H00を透明色に固定し、背景の基本を透明色にしていること（画面の重ね合わせに便利）などなど。

特に背景が透明色だというのは嬉しい。どこでどう間違ったのか、絵は白い画面に描くもの、といった重力に魂を引かれたソフトが多いからだ。

まず、ファイル入出力モードで立ち上がる。3画面+コピーバッファ、そして6つのパレットに入れたいファイルがあったら読み込むのである。終了時もこの画面に出て、セーブするなりする。デザインはともよい。

そこからコマンドシェルを起動することもできる。drawを選ぶと縦長でかくてデザインを優先したようなポップなウィンドウが現れる。アイコンがたくさん並んでいて、カラフル。ウィンドウは3つに分かれており、上1/3が描いたりコピーしたりするもの。まん中がパレット。その下がパレット関係の処理。

たとえば、グラデーションバーの両端に



絵の一部をペンとして使う

色をセットして、そのあいだの色の变化パターンをいくつにするか決める。それでも、パレット上のそのグラデーションをセットしたいところへ置くと、ずらっとグラデーションした色がパレットに置かれるのである。マジックパレットというグラデーションは、あるパレット番号からあるパレット番号への色の並びにすぎないので(中身の色はなんでもいい)、赤黒青緑といった4段階グラデーションもできるし、虹も描ける。

処理の基本は前にも書いたが、カット&ペーストである。任意領域をカットしてバッファへ移し、それを任意の位置へペーストする。バッファを編集したりできるし、透明色を背景にしておくと、重ね合わせが簡単にできる。

その代わり、縮小・回転・変形処理が任意の領域に対してできない。回転や縮小をしたいときは、対象のものだけをほかの画面へ持っていき、画面全体の256×256の画面に対して行ってから、戻すといった作業が必要で、複雑な絵を描こうと思ったら、まめにパーツをセーブするのがいいだろう。あと、トーン処理も面倒だ。

アンドウ処理はユニーク。アンドウはアンドウ用画面メモリから戻されるのだが、そのメモリへのデータ格納は手動なのだ。で、面白いことに、消しゴムを使って画面を消すと、その下にはアンドウ用メモリの画像が現れるのであった。アンドウというより、いろんな画面効果に使えそうだ。

無理をいえば、任意のパレットを使ったカラーイメージユニットからの取り込みか、もっと上手な65536→256色変換がほしい。メニューやコピー時の領域が画面内に制限されているので画面が狭く、ちょっと不便なものも惜しいところだ。

なんだかんだいっても、Z'sSTAFFの影響を免れないソフトが多いなか、こいつだけは違う。非常にポップで軽く遊ぶには最適だ。

16色モードでは画面上の絵をスプライトデータに落とすことも可能だし、起動時にSキーを押しながら立ち上げると直前に走っていたゲームなどのスプライトデータとスプライトパレットを読み込んでくれるのでスプライトエディタとしても使える。

おまけで、マジックパレットのデータをBASICで使うための関数やBASICプログラムのサンプルがついてきて、とても便利である。ついでに、Cのライブラリがあればコンパイルできてよかったのに。オートデモもある。



太めのペンでベタベタと描いてみた。油絵調に見えるかな？(河○純子ちゃんがモデル)マウスボタンの左右に色を設定でき細かい修正に便利。マウスの反応速度を調整できるのもいい。特殊効果に弱いのとスキャナ・プリンタに対応していないのが辛い。

PRISM

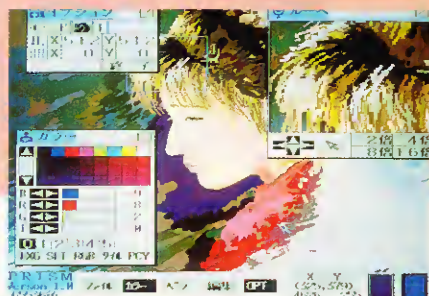
こいつはZ'sSTAFFの影響を逃れられなかった。最初から大樹の陰にいたのかもしれない。

ウリは、2色から65536色まで、256ドットから512ドットまで対応した多彩なモードと、アニメーション機能。さすがウルフ・チームである(そーいえば、昔侍ジャイアンツにウルフチーフって選手がいたなあ)。

しかし、なんといっても、円が描けないとかグラデーションペイントができないとか文字入力がないとかペン先やブラシの編集もできないとかカラーイメージユニットもイメージスキャナも使えないといった事情にはなにか深いわけでも……と考えてしまう。

その他の操作性は遅いZ'sSTAFFという感じがた。ウィンドウデザインも似ている。

ウリはやはりアニメーション機能か。画面上の任意の矩形をたくさん切り出して、連続して見せてアニメーションしてしまうという機能だ。まずマウスで1コマの大きさを決め、15コマまで任意の位置を切りとって並べる。1/60秒単位で1コマの時間を指定できるから、サブリミナル効果測定テストなんかもできて面白いぞ。



どのタイミングが一番いいかテストして、学園祭では売り上げ倍増だ！(そんなにうまくはいくもんか)

画面を2画面まで持てるので片方を背景に使うとかすれば、なかなか、このソフトの意図も見えてくるかもしれない。

ゲームでは特殊な画面モードを使ったりするためか、ふつうのグラフィックツールではサポートしないような512×256ドットモードなどや16色モードなどにも使えるが、その半面、どのモードでもできるような機能しかついていないのが残念だ。とりあえず、どのモードでも絵は描けることを特徴としている。

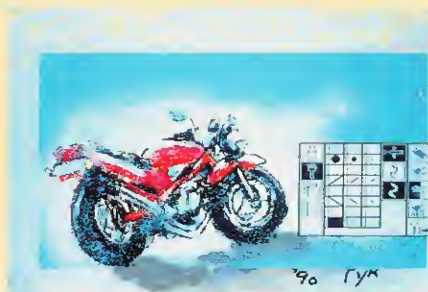
256色モードでアニメーションして遊びたい人は、マジックパレットとPRISMの2つを買って、マジックパレットで描いてPRISMで動かす、ってのもいいかもしれない。定価ベースでは、この2つを買ってもまだZ'sSTAFF PRO-68Kより安いのだ。



PRISMの使用例。油絵調を狙ってみた



手元にあった鉛筆の落書きを、ハンディスキヤナで取り込んでエディットした。水彩画を意識してみたが……。ウィンドウ操作が比較的速いので、カラーチェンジや閉曲線コピーなど編集機能が多く揃っているのがよい。ただし入出力ファイルの形式がGL3なのが少々不満。ルーペは画面一杯に拡大し、そのままエディットできる。



G68K II Version 2.0-PRO

バージョン1では、日本初のBGMつきグラフィックツールという快挙を成し遂げてくれたG68Kであるが、バージョン2ではおとなしい作り（というかまともな作り）になっている。

立ち上げて驚くのが、真っ白な画面にボツンと十字カーソルがあるだけのまぶしい画面。メニューバーからメニュー選択するプルダウン式ではなく、その都度右ボタンでメニューを開いていくポップアップ式なのだ。

たかがツールされどツール

CGコンテストの審査などでよくいわれることですが、応募されてくるものに「こんな機能を使ってみました」みたいな作品が結構多いのです。ツールの豊富な機能を使うのはいいのです。でも、それに振り回されて自分の表現したいものがあやふやになっては駄目ですね。作品はツールの機能紹介ではないのですから、饒舌すぎないオリジナリティのある作品を描いてもらいたいです。

それには自分にあったツールを深すことも必要でしょうし、最終的には自分自身で組んだ、自分のためのお絵描きツールを使うのがベストなんじゃないかな。

昔、(PC-9801の話だけど) Z'sSTAFFと並んで有名だったグラフィックツールにシステムソフトのアートマスターというのがあった。このアートマスターもポップアップメニューで、アイコンやらメニュー構造などが非常に似ている。要は慣れの問題で、開いたウィンドウがうっとうしいという人もいれば、いちいち右ボタンでウィンドウを開くのがうっとうしいという人もいる。

機能的にはグラデーションペイントがないくらいで、普通。

パレットにタイル模様もセットできたりとか、マスク機能が使いやすいといった長所もある。使い勝手の差は、ポップアップメニューが馴染むか否かだろう。

独自のファイル構造や圧縮方式を持って

パソコン通信を始めてから、PC-9801で描いたイラストを見る機会が多くなりました。うまい人の絵を見ると、16色という限定された色数を巧みに利用してとても美しい効果を表現しています。レイトレーシングや取り込み画像など特殊な用途以外なら、多色よりむしろ少色のほうがセンスのある色彩設計ができるのではないのでしょうか。

ところで16色768×512モードのCGツールってありませんねえ。あればPC-9801の絵を利用しやすくなるんですけど。どっかでエスキースみたいな16色CGツール出ませんかええ。やっぱり自分で作るしかないのかなあ。(T.F.)

おらず、データはすべてGL3形式というのが素直といえば素直でよい。

Z'sSTAFFをよほど意識しているらしく、Z'sSTAFFの非圧縮ZIM形式とGL3形式の相互ファイル変換が可能となっている。

機能的にはZ'sSTAFFと比べるのがかわいそうだが、価格が半分以上であること、Z'sSTAFFより少ないメモリで動くといった面もあり、一概にはいえない。

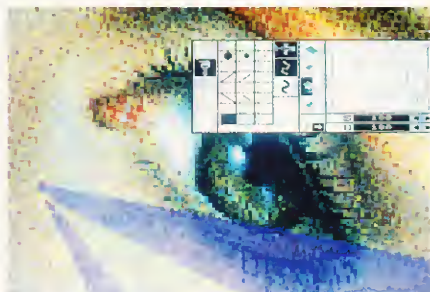
綺麗なサンプルとオートデモあり。

まだまだ、先はあるのである

X68000のグラフィックツールといえば、多くのユーザーや開発者がZ'sSTAFFを基準にしてきた。それはそれでいいとして、Z'sSTAFFが完璧なソフトか？ というと、決してそんなことはないのである。そのひとつの例をマジックパレットが証明したわけだが、まだまだいろいろ便利な機能はあるはずである。マジックパレットだって、早く次のバージョンを！ てな感じだ。画像取り込みの柔軟さと、任意矩形の変形はほしいところ。グラデーションなんて簡単に綺麗な効果が出せるだけで乱用すると見苦しいだけだし。

えっと、コンピュータを使って絵を描くことの意義を考えてもらいたい。絵心のあつた人がペンをマウスに持ち替えて、ああ、よかったね、という時代は過ぎ去った。わざわざマウスを持たせるのだから、結局ペンで絵の描ける人でないといひこなせない、というのは変である。Z'sTRIPHONYやC-TRACEなどはデッサン力がなくても、センスと根性と待つだけの暇とちょっとした頭があれば誰でも使えるものだった。2Dグラフィックツールも、そんなものが出てきてもいいではないか。

たとえば、遠近法矩形や、始点と終点で太さの変えられるペン。任意の方向へのグラデーション。多彩なアンドウや下書きプレーン（メモリの関係で大変だろうけど）。別に65536色でなくとも、32768色でも16384色でもいいので、そういった支援機能を充



メニューは邪魔にならないポップアップ式

ギザギザのないグラフィック関数

アンチエイリアスとは？

Tan Akihiko 丹 明彦

というわけで、2次元グラフィックである。これまでは3次元グラフィックが主だったので、次元がひとつ落ちたことになるのだが、それはつまり、質的にも一段と落ちたことなのか？ いやいやとんでもない。2次元のほうが3次元よりもずっと身近で表現しやすいのである。そして表現しやすいぶん、人は精魂こめて絵を作り上げるし、質的にも高いものができる。そのことはOh!X誌に毎日のように送られてくるイラストを見てもわかる。とにかく、層の厚さが違うぶん、競争も激しいし、いいものしか残らない。これはとてもいいことである。

さて今回の目標は

これから紹介するのは、コンピュータのスクリーン上によりよい2次元の1枚絵を作るための道具である。といってもX-BASICのグラフィック関数とやっていることは基本的には同じ。1つひとつの関数の動作は非常にプリミティブなもので、現段階では「ペンと紙とスクリーンとペンをキーボードに持ち換えた」と同じような感覚で使うことは、残念ながらまず無理である。優れたグラフィックツールであるZ's STAFFでさえ、ただペンをマウスに持ち換えただけなのとは少し違うのだが、それとは次元が違う。

今回制作しようというX-BASICの外部関数は、マウスから入力するといったユーザーインタフェースについては無視である。つまりその部分はユーザーであるあなたにお任せ、ということになる。用意したのはやや強力なラインやペイントなのだから、それをあなたがどう活用しようとまったく自由である。

X68000でラインやペイントを使った2次元グラフィックで良質なイラストを作ろうというのが今回の試みだといったが、こういう反論もあるだろう。「X-BASICにだってラインやペイントはあるぞ、どうしていまさら作り直す必要がある？」と。そ

の考えは甘い。X68000の標準グラフィック関数は、せっかくの65536色を生かしていないのである。

コンピュータで描いたイラストの多くがどうして雑誌の表紙を飾りうるだけのクオリティを持ちえないのか。よくできてはいるけどどこか違和感のあるイラスト。そのひとつの解答がここにある。輪郭に出てくる見苦しい階段、すなわち「ジャギー」である。

'90年のトレンドはドッター

その昔、人間デジタイザと呼ばれる人々がいて、変な奴と思われながらも尊敬を集めていた。かれらの道具はラインとペイントであった。当時はマウスなどという便利な道具は庶民の持つべからざるものであった。Z's STAFFのような操作が簡単なうえに強力な表現力を持ったグラフィックツールに至っては、夢のまた夢であった。

そこで彼らは方眼紙に下絵を描き、座標値を丹念に取りながらぼちぼちとキーボードから打ち込んでいたのであった。そしてラインで線を引き、中をペイントで塗りつぶす。

いまでこそ総天然色（ちと古いか）は常識でも、8色が主流であった時代のこと、微妙な筆づかいなどは表現しえようはずもない。そのため古来の名画を模写するような試みはあまりなく、彼らの興味はもっぱらアニメ絵に向いていた。パソコン使いとアニメファンの深い関係はこうしてできあがったのであろうか。

そして時代はアナログに向かい、高品質の絵を誰でも作れる、そんな期待を感じさせるマシンの登場を見た。X68000である。ところがその期待はまだ期待の域を出てはいないのかもしれない。

X-BASICでline()関数を使ってみた方は、およそ滑らかさが無いのに驚かれたのではないかと思う。もちろん、従来機種ではそれが当たり前のことだったのだが、せ

コンピュータグラフィックでの強敵のひとつジャギー。今回はこれを追放すべく、新しいグラフィック関数を揃えてみました。もっとエレガントなラインルーチンと高機能なペイントルーチンなどによる高画質な2Dグラフィックワールドを構築していきます。

めて65536色モードのときくらい、もっと目に優しい線がほしい……というわけでZ's STAFFに期待がかかるわけであるが、こちらでも残念ながら完全なサポートはなかった。

この件の解決法はいくつかある。

- ・中心部が濃く周りが薄いペンを指定して、ふつうに線を描く
- ・ただの線を描いてあとからぼかす
- ・あきらめる

3番目は問題外として、どれも自然で滑らかな線にはならない。さらに共通の欠点もある。これらの方法でそれらしく見えるように線を描けたとしよう。すると、こういう線で囲んだ内側をペイントしようとしてずっこけることになるのである。ペイントできないのである。いや、できることはできるがきちんと隅々まで塗りつぶしてくれないのである（手元にZ's STAFFのある方はお試しいただきたい）。というわけで最後の手段として、

- ・ルーペで拡大して1点1点描く
- ということになるのである。

現在あちこちで（市販ゲームのビジュアルシーンなどで）見かける比較的良質な画像のほとんどは、こうやって描かれている。現在のデジタルペインティング界を支えているのは、このドッターたちなのだ。

僕はこのルーペでドット打ちという作業を自分ではしたことがないので、はなはだ無責任な意見ではあるのだが、どう見ても非人間的な作業としか思えない。この点、人間デジタイザと似通っている。

しかし描いている本人は決してそうは思っていないであろう。この手の作業は慣れると苦しくはなくなるものである。それに つれて質も上がってくる。しかしどうしても職人芸になりがちである。いきおい選ばれし者の技術になってしまう。そして一般ユーザーからは変な奴とか閑人とかのレッテルを貼られてしまうのである。合掌。

今回はそこまでの質を追求するつもりはない。BASICから手軽に使えればよい。い

ろいろと遊べたらなおよい。そんな気持ちで作ってみた。

アンチエリアシング

で、さきほどもちらっと出てきた「ジャギー」である。これは昔から再三いっているように、有限個しかないグラフィック画面のドットで、無限といってもいい細かさの画像を表現しようという要求のなかで、起こりうるべくして起こる問題である。サンプリング理論の言葉で「エリアシング」という。

これを防ぐためには、視力の限界を超えた高い解像度のCRTを使うのが完璧な解決法であろう。しかしそんなものはないし、あっても化け物のように高価であろう。

ではどうするのか。うまいことフィルタをかけて、不連続に変化しそうなピクセルの輝度の変化を補間するというのが現在もっとも効果を上げている方法である。

黒い線を引いたつもりでも、その緑の部分には微妙に灰色のピクセルが並んでいて、遠くから見れば滑らかな線に見えるのである。境界をぼかしてごまかしているのと混同されがちだが、これはぼかし処理とはまったく異なるもので、アンチエリアシングと呼ばれる。

百聞は一見にしかずというわけで、まずはなにもいわずにリスト1を実行していただきたい。いうまでもなくX-BASICのリストである。

図1 点列のデータ構造

dim int pts(N, 2)で定義する。Nは点列のおおよそのサイズ。

N	0	1	2	
0	n	0/1	0/8	ヘッダ
1	x ₁	y ₁	w ₁	頂点の情報
2	x ₂	y ₂	w ₂	
3	x ₃	y ₃	w ₃	
⋮	⋮	⋮	⋮	
n-1	x _{n-1}	y _{n-1}	w _{n-1}	
n	x _n	y _n	w _n	未使用
n+1				
⋮	⋮	⋮	⋮	
N				

pts(i, j)の意味

[ヘッダについて] (i = 0)

pts(0, 0) = n
pts(0, 1) = 0または1

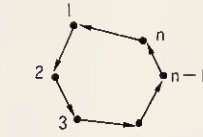
頂点数(≤ N)
点列のタイプ



pts(0, 1) = 0
多角形は開いている
pts(0, 2) = 0または8

[頂点について] (1 ≤ i ≤ n)

(x_i, y_i)
w_i



pts(0, 1) = 1
多角形は閉じている

オーバーサンプリング座標のフラグ
オーバーサンプリング処理がされていない
オーバーサンプリング処理済み

第i点の座標
第i点での線の太さ(aa_lines()関数)

線が4本画面に見える。そのうちいちばん上といちばん下の線は画面の中央でがたんと1ドット上がり、2本の線を1ドットずらして継ぎ合わせたような印象である。1本の線には見えない。

対して内側の2本はわりと綺麗な線に見える。そしてこの4本は、全体としては平行線である。右端と左端を見ると、確かに等間隔である。とすると、内側の2本はまっすぐな線に見えるようだが、状況から見て、どこかで1ドット上がっていかなくては辻褄が合わない。

ここでZ'sSTAFFをお持ちの方は、ルーペを使って、内側の2本がどのような色使いをしているのかを見れば、アンチエリアシングの原理がおぼろげにでもわかることであろう。

しかしこれではあまりにも応用がきかない。今回作ったのは、もっといろいろな線にも使えるようなプログラムである。その具体的なアルゴリズムの説明はあと回しとしよう。とりあえず使えることが大切だ。

使い方である

どういう形式で実現するか迷ったのだが、手軽に使えるX-BASIC外部関数という線に落ち着いた。その関数本体はC言語で書いている。

X-BASICの外部関数をCで書くときの注意や、コンパイルの手順などは困みにしであるのでそちらをご覧くださいとして、

いま、あなたの手元には今回作った外部関数anti.fncを組み込んだX-BASICが起動しているものと思って話を続けることにする。

関数のリファレンスマニュアルを表1に掲げる。anti.fncにはこの表にない関数も収録してあるが、隠し関数のようなものなので、とりあえず表1に載っているものだけを使っていたきたい。

anti.fncを使いこなすには、表1や図1にも出てきている「点列」というデータ構造の把握が不可欠である。というよりもそれがほとんどすべてである(点列にはCのソースファイル中ではPTSという型を与えている。pointsを略して命名した)。

点列の基本単位は整数3個で、それが(頂点数+1)個並ぶかたちになる。X-BASIC上では、

```
int pts(10, 2)
```

と宣言する。BASICの配列の宣言は、Cのそれと少し違っていて、同じ宣言をCでは、

```
int pts[11][3];
```

リスト1

```
10 /* アンチエリアシングの原理
20 /* 1ドットの段差を持つ線
30 screen 1,3,1,1
40 /* アンチエリアシング
50 for x=0 to 511
60   i=32*x/512
70   pset( x,199,rgb(i,i,i) )
80   pset( x,200,rgb(31-i,31-i,31-i) )
90   pset( x,204,rgb(i,i,i) )
100  pset( x,205,rgb(31-i,31-i,31-i) )
110 next
120 /* ノンアンチエリアシング
130 line( 0,210,511,209,65534 )
140 line( 0,195,511,194,65534 )
```


とする。BASICは添え字の最大値を、Cは1次元あたりの要素の数を基本にしているからだが、あとの参照や代入のしかたは両者ではほとんど変わらない。

点列の宣言は前述のとおり2次元配列で行うが、第1添え字(10)は頂点の数の最大値というか、その目安を適当に決めて設定する。たとえば複雑な形なら値を大きくする。曲線を記録する(後述)ときも大きくする。第2添え字のほうは2に固定である。

点列の構造について少し解説しよう(図1)。頭から3要素、pts(0,j)は少し特殊

で、ヘッダと呼んでいる。pts(0,0)には実際の頂点数が、pts(0,1)には点列のタイプが、pts(0,2)にはオーバーサンプリング倍数が入る。

点列のタイプは2つに分かれる。それを理解する助けとして、1本の紐を想像してもらいたい。その紐が点列を表している。いま、その紐の端と端を結んだとする。その状態が、点列タイプ=1の状態、循環していると名づける。要するに閉じているわけである。そうでない、開いている状態が点列タイプ=0というわけである。

今回のプログラムの作り方

X-BASICの外部関数をC言語で作るわけであるが、今回のプログラムは、

- ・内蔵の関数(機能)が比較的多い
- ・それぞれの処理が多少複雑
- ・したがってプログラムサイズが大きい
- ・たったひとつの関数をデバッグするのにいちいち全部コンパイルしなおしてはやりきれない

というわけで、分割民営化、じゃない、

分割コンパイル

の採用に踏み切った。複数のソースファイルを別々にコンパイルして、最後にリンクを使ってひとつにまとめるやり方のことである。僕も今回ほどバラバラにしたのは初めてだが、いざやってみると非常に快適である。

0) 環境

最初に開発環境を確認しておこう。

使うCコンパイラはXCかGCC。コンパイラはどこに置いておいてもいいが、パスは通っていないてはならない。コンパイラのほかにもアセンブラ(as.x)とリンクラ(lk.x)が必要である。これらにもパスを通しておくこと。当然ながらテキストエディタも必要。僕はmicroEmacsを使っているが、標準的なのはed.xであろう。

設定しておかなくてはならない環境変数もいくつかある。autoexec.batなどに次の設定がされているかどうか確認しておくこと。システムがAドライブでRAMディスクがFドライブの場合、

```
TEMP F:
SET lib=A:\LIB
SET include=A:\INCLUDE
BASICの入っているディレクトリは、
A:\BASIC2¥
```

とする。そうでない方は各自のシステムにあわせて読み換えてほしい。

ほかに大切なのはインクルードファイル(*.h)およびCのライブラリ(*.lib.a)である。それぞれ、

```
A:\INCLUDE¥
A:\LIB¥
```

に収めておくこと。C compiler PRO-68Kのシステムディスクの設定なら基本的には安心してよい。そうそう、GCCの場合は、gnulib.aというライブラリもあるが、これもA:\LIB¥に収めておけばよい。

1) ソースリスト作成

環境設定ができたから、さっそくソースリストを作ろう。打ち込むリストは次のとおり。すべてつづきにテキストエディタで打ち込む。

・anti.s (外部関数ヘッダ)

・anti.h (マクロ定義ファイル)
・main.c (引数リスト宣言)
・pts_curve.c (自由曲線の発生)
・pts_procs.c (輪郭の処理)
・aa_lines.c (輪郭描画)
・aa_scanconv.c (多角形塗りつぶし)
・aa_paint.c (閉領域ペイント)
・aa_procs.c (タイル・トーン処理)

一度に全部打ち込むのもおっくうなので、テストしながら作業を進めたい方や、必要ない関数を打ち込みたくない方は、そういう関数の名前だけ書いて中身を書かない(return(0)だけは入れておいたほうが安全だが)という手が使えるので参考にしていきたい。

2) コンパイルおよびアセンブル

ソースリストを打ち込んだら、それぞれをコンパイルする。ただし、anti.sとanti.hは例外。anti.sはアセンブラ(as.x)でアセンブルする。

```
as /u anti.s
```

エラーがなければ、anti.oというファイルができる。anti.hのほうはただのインクルードファイル(それぞれのCのソースにインクルードして使う)で、それ自身を単独でコンパイルする必要はない(してもなにもできてこない)。

さて、Cのプログラムのコンパイルだが、こちらもふつうどおりではない。分割コンパイルなので、リンクフェイズまで一気に突っ走ってはいけない。～.oの段階で止め、最後にリンクするのが分割コンパイルである。だからリンクフェイズの直前でコンパイルをやめるスイッチをコンパイラに与えてやらなくてはならない。これがXCとGCCでは違って、それぞれ、

```
cc /L ~.c
gcc -c ~.c
```

である。また、GCCの場合は最適化オプションが豊富なので、それもついでに与えよう。いちいち長たらしいオプションを打ち込むのは面倒なので、次のようなバッチファイルを作ることとする。これもテキストエディタで書く。ファイル名は仮にcompile.batとしよう。

(XCの場合)

```
cc /L %1.c
```

(GCCの場合)

```
gcc -c -O -fstrength-reduce -fomit-frame-pointer -finline-functions %1.c
```

バッチファイルができたなら、

```
compile main
compile pts_curve
:
compile aa_procs
```

オーバーサンプリングについては、もう少し後ろで説明するが、予備知識として簡単にいっておくと、今回の目玉であるアンチエイリアシングに使う技法である。

ひとつのピクセルをより細かく分割して図形を描き、出力する段階で平均すれば、最終的に出てくる図形の輪郭が滑らかになるという思想に基づいている。座標系を、ピクセルのサンプリング周波数よりもっと細かく取るから、オーバーサンプリングと呼んでいる。今回は8倍オーバーサンプリングとしたので、pts(0,2)には0か8が入る。

と各ソースごとにコマンドラインから実行する。もしエラーが発生したりバグを取ったりしたファイルがあれば、そのファイルだけをコンパイルしなおせばよい。

3) リンク

ここまで無事終了したら、～.oというファイルが8つできていることであろう。そこで仕上げのリンクフェイズ。

```
lk /o anti.fnc anti.o main.o pts_curve.o pts
```

```
_procs.o aa_lines.o aa_scanconv.o aa_paint.o
aa_procs.o %lib%¥clib.a (%lib%¥gnulib.a) %lib
%baslib.a
```

カッコ内のgnulib.aというのはGCC専用のライブラリで、いうまでもなくXCでコンパイルする人には必要ない。/oオプションを使って、ふつうならanti.xとなる出力ファイルの名前を外部関数の名前anti.fncにする。実はX-BASICの外部関数の正体は実行形式ファイルと同じである。ただ名前がそうになっていないだけ。

4) インストール

あとはX-BASICにできたのanti.fncを組み込むだけである。まずBASICのディレクトリにanti.fncを転送する。

```
copy anti.fnc A:\BASIC2¥
```

それからBASICのディレクトリ上のコンフィギュレーションファイルをテキストエディタで書き換える。標準ではbasic.cnfというファイル名である(X-BASICは/cオプションを使って指定したコンフィギュレーションファイルで立ち上げることもできる)。

以下はその一例である。大切なのは最後の1行。

```
FREE = 128
WIDTH = 64
BEEP = ON
CAPS = OFF
FUNC = GRAPH
FUNC = PIC
FUNC = ANTI
```

ほかにも音楽関係の外部関数を組み込んでおけば、音楽を演奏しながら絵を描くという芸当もできるだろう(してなんになる)。ところで下から2行目のpic.fncというのは、やはり本誌6月号の付録ディスクについていたPIC形式の画像ファイルをセーブ/ロードする外部関数。描画の遅いanti.fncにとってはとてもありがたい相棒である。

これでやっと使えるところまでこぎつけた。正直いって、Cとアセンブラを扱いなれた人にはこんな説明は退屈なだけかもしれない。

pts(i,j)は、 $1 \leq i \leq \text{pts}(0,0)$ である i については i 番目の頂点の情報を格納する。pts(i,0)にはx座標が、pts(i,1)にはy座標が、pts(i,2)には線の幅がそれぞれ入る。

それでは動作チェックも兼ねて簡単な使い方を練習しよう。まずは点列の宣言の方法から。

例1) V字型

```
dim int p1(3,2) = { 3,0,0
,100,100,8
,200,300,8
,300,100,8 }
```

例2) 三角形

```
dim int p2(3,2) = { 3,1,0
,100,100,8
,200,300,8
,300,100,8 }
```

この2つのサンプルのあいだでは、点列タイプ (pts(0,1)) だけが違うことに注意しよう。

点列の定義ができたら、それを使ってなにか描いてみよう。その前に、完全に制作者(要するに僕)の都合なのであるが、点列をオーバーサンプリング座標系に変換しなくてはならない。変換をかけておかないと、この先出てくるほとんどの関数が使えない。ま、ここはおまじないとも思っておこう。

```
pts__oversample(p1)
pts__oversample(p2)
```

次に、画面モードを65536色モードに変える。ちょっと手抜きなことに、描画関数の中に画面モードのチェックを入れていないので、忘れずに実行しておくこと。

```
screen 1,3,1,1
```

では先ほど作った三角形を画面に出してみよう。

```
aa__lines(p2, rgb(31,31,31))
```

なかなかダルイが、おしまいまで待とう。白い三角形が出てくると思う。

お次はいまの三角形の頂点を通る曲線を作ってみよう。それにはまず、曲線を格納する配列をひとつ用意する。というのも、曲線は短い線分をたくさんつなげてそれらしく見せるようにしているからだ。そのため、ある程度多くの頂点も記録できるように大きな配列を用意する。余裕を持って、

```
dim int p3(1000,2)
```

と大きめに宣言しておき、すかさず、

```
pts__curve(p2,10,10,p3)
```

を実行。

pts__curve()は曲線を生成するだけの関数なので、画面にはなにも出ないはずだ。ちょっとしたら戻ってくるので、できた曲

線を見てみよう。さっきと同様に、

```
aa__lines(p3,rgb(31,0,0))
```

今度は赤い色で三角形のカドを取ったような曲線が出てくるはず。

さてここでいったんご破算願おう。

```
wipe()
```

そして新しい気持ちでもっと妙な形を試してみることにする。

```
dim int p4(6,2) = { 6,1,0
```

```
,100,100,8
,200,300,8
,300,100,8
,400,400,8
,300,200,8
,200,400,8 }
```

例によってオーバーサンプリング座標に変換するおまじない。

```
pts__oversample(p4)
```

この「N」をひっくり返したような多角形の頂点を通る曲線を作る。

```
dim int p5(2000,2)
```

```
pts__curve(p4,8,8,p5)
```

さっきは輪郭線だけだったが、今回はこの曲線の内側も塗りつぶしながら描く。

```
aa__scanconv(p5,0,65534,0,0)
```

白い変な形が現れる。その中を赤でペイントしてみよう。

```
aa__paint(250,250,0,rgb(31,0,0),0,0)
```

ちなみに、このaa__paint()の代わりに、paint(250,250,rgb(31,0,0))

を実行してみると、aa__paint()がアンチエイリアシングに対応しているありがたいペイント関数であることがわかることだろう。

以上の動作に支障がなければ、ほぼバグはないと考えていいだろう。表1の関数リファレンスを参照しながら、上の例題の数値をあちこちいじって実行してみよう。そして、それぞれの関数がどういう機能を持ち、どんなパラメータを与えるとどんな動作をするか、そういうことを理解して、さらに難しい作品へと進んでいってほしい。

アルゴリズム解説

ソースリストが思ったより大きくなってしまい、我ながら驚いている。こんなものの説明をすることを考えるだけで胸焼けである。ま、すべてはソースリストが語ってくれるということで、コーディングするうえでの細かい注意は、ソースリストに入れたコメントに頼ることにし、ここではアルゴリズムの心を語ることにする。

今回の外部関数を構成するための主要なアルゴリズムはいくつかある。幸いなこと



アンチエイリアシングの奇跡

に、過去のOh!X誌ですでに僕が紹介しているものも多いので、適宜参照していただきたい。

オーバーサンプリング

アンチエイリアシング技法のなかでもっともポピュラーな方法のひとつが、このオーバーサンプリングである。レイトレーシングやZバッファといった3次元CG技術をアンチエイリアシング対応にする場合、必ずといっていいほど用いられるのもオーバーサンプリング。

ここまでの説明でもちらっと触れているのだが、まず事実として、ピクセルのサンプリングレート(要するに解像度)はかなり高いように見えて、人間の目をごまかしおおせるほどには高くないということがあ。そこで多色表示の利点を生かすことが考えられた。ともすれば急激で不連続的になりがちなピクセルの輝度変化をもっと滑らかにし、曖昧な(少し語弊があるが)輪郭を作れば、目に優しい画像ができあがる。

そのために、いったんピクセルよりも高いサンプリングレートで画像を生成しておく。このときの最小の処理単位は、ピクセルよりもさらに小さな画素であり、サブピクセルと呼ばれる。

ちなみに1本のスキャンラインも数本のさらに細いスキャンラインに細分されることになり、サブスキャンラインと呼ばれる。今回の外部関数では8倍オーバーサンプリングを採用している。この場合1ピクセルは $8 \times 8 = 64$ サブピクセルからなる。

描画アルゴリズムは従来の(オーバーサンプリングを用いない)アルゴリズムを拡張して使う。ただ処理単位がピクセルでなくサブピクセルになっているだけである。

そして、1ピクセル中の全サブピクセルの輝度を平均してスクリーンに出力すれば、粗いピクセルにそれ以上の解像度を持たせたのと同等の効果が得られるという仕掛け

になっている。

誤解を恐れずにいうなら、アンチエイアシングは人間の目を巧みにごまかす技法であるともいえる。もちろん、ピクセルをよく見ればそんなごまかしはすぐわかってしまうし、1ピクセルを下回るような細かい図形には効果が薄くなってしまうといった欠点はあるものの、いたずらに解像度を上げるよりもずっといい方法なのである。

今回の描画アルゴリズムでは、サブピクセルの輝度を1つひとつ配列に持っておくことはしなかった。2次元なので、基本的に隠面処理など考える必要はない^(*)、それならば「いま描画しようとしている図形が各ピクセルのうちいくつのサブピクセルを占めているか」という情報だけが重要だとわかる。これをピクセルあたりの寄与率と呼ぼう。以後は α という記号を使うことにする。

8倍オーバーサンプリングの場合、サブピクセル数は0から64の値をとる。 α はこれを64で割った値、つまり $0 \leq \alpha \leq 1$ の間の値をとる。ピクセルと図形がまったく重ならない場合は $\alpha = 0$ だし、ピクセルを図形が全部覆っている場合は $\alpha = 1$ 。境界部だけで α は0でも1でもないいろいろな値をとる。

α は一般に実数だが、プログラム上は実数よりも整数のほうが取り扱いが楽なので、ひとまず $0 \leq \alpha' \leq 64$ で格納しておき、最後に64で割っている。これでも結局は同じである。

スクリーン出力の段階では、 α 合成と呼ぶ方法を用いる。背景が真っ黒な場合は α がそのまま輝度になるのだろうが、もちろんいつでもそんなことがあるはずはなく、ふつうは、適当な比で図形の色と背景の色

を合成しないと、輪郭が変になってしまう。この比に α を用いるのである。つまり次の比で混合する。

図形の色：背景の色 = $\alpha : 1 - \alpha$
参考) この方法の画質をもっと上げる方法として、重みづけ平均化をすることも考えられる。 α を出す段階で、ピクセルの中心部のサブピクセルほど α に大きく寄与するようにプログラムを組んでおくのである。今回採ったのは単純平均化で、どのサブピクセルも同じ重みをもっていることになっている。

Bresenhamのアルゴリズム

昨年解説したZバッファアルゴリズムの前フリとして線分描画を説明した(1989年7月号)。一般に線分の傾きは実数である。実数である線分の傾きを相手にしながらも、Bresenhamアルゴリズムはすべての演算を整数ですませてしまう。このアルゴリズムは、実際に応用が広い。たとえば本誌5月号のグラフィック拡大縮小にも使っている。

Bresenhamアルゴリズムの核となる部分を以下に示す。 $(x1, y1)$ から $(x2, y2)$ へ色 c で線分を引く。ただしここでは $x1 < x2, y1 < y2$ である。ほかの場合についてもそれほど難しくない拡張で対応できる。

Bresenhamアルゴリズムの基本的な考え方は、ピクセルの中心と真の線分との上下関係を比べ、真の線分にもっとも近いピクセルを点灯していくというだけのことである。この上下関係を比べるのに、誤差と呼ぶ量 e を使って処理を効率的にしている。

$dx = (x2 - x1);$
 $dy = (y2 - y1);$
 $e = -dy;$ (誤差の初期値)

```
for (x=x1,y=y1;x<=x2;x++) {  
    pset (x,y,c); (ピクセル点灯)  
    e += (2*dy);  
    (1ピクセルあたりの真の線分の上昇分)  
    while (e>=0) {  
        (真の線分が上にあるあいだは)  
        y++;  
        (ピクセルの座標を上げる)  
        e -= (2*dx);  
        (その分だけ真の線分との距離を詰める)  
    }  
}
```

もっと詳しく知りたい方は1989年7月号の記事を参照してほしい。

ライン

ただの線分ならば上のBresenhamアルゴリズムを使うのだが、アンチエイアシング対応となるとそう簡単にはいかない。しかも今回は欲張って、線分の幅を変えられるようにしたのでよけい厄介である。

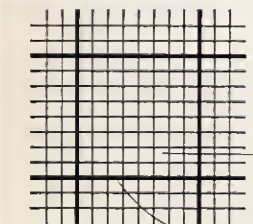
それでは、 $(x1, y1)$ から $(x2, y2)$ へ幅 w で線分を引くことを考えよう。といってもそれほど難しいことではない。まず描きたい太い線分を1ピクセル間隔で切る。イメージとしては輪切りである。そしてそのひとつひとつ(幅1ピクセルで長さ h ピクセルの小線分)をスクリーンに張り付けるのである。切り口の長さ h は、ピタゴラスの定理(おお懐かしい)を使って求めることができる。

ここまでわかればあとは簡単。まず太い線分の下端(これも線分になる)を通常の線分と同じようにBresenhamアルゴリズムで発生させる。

具体的には $(x1, y1 - h/2)$ と $(x2, y2 - h/2)$ を結ぶ線分、すなわち太い線分の中心から h の半分だけ下にずれた線分である。そして、この下端の線分の上に長さ h の小線分を並べていけばいい。これは、まっとうに描けば傾いた長方形になるはずの太い線分を、平行四辺形で近似したことになる。あまり線分が太くないうちはたいして不都合はおきないが、太くなってくると不自然さも目立つし、ときには破綻することもある。

(*) 3次元CGでござますがにこんないい加減なことではすまされず、きちんとサブピクセル数だけ α を $1/256$ を単位として用意し、隠面処理をきちんと終えてから合成するという手順が要求される。これはあまりに原始的な話で、実現するうえではもっと効率的な方法も提案されている。

図2 オーバーサンプリング



ひとつのピクセルを、それより細かい仮想的なサブピクセルに分ける。
たとえば左の図は8倍オーバーサンプリングで、ひとつのピクセルは $8 \times 8 = 64$ 個のサブピクセルに分かれる。

サブピクセル

ピクセル

左のような図形の、各ピクセルへの寄与率は次のようになっている。

21	59
64	64
32	64
64	64

この寄与率 α からピクセルの輝度を求め、出力する。



寄与率にしたがってピクセルの明るさを変えてアンチエイアシングを実現する。

る。このことはあとで触れる。

いずれにせよこれで太い線分は描ける。あとはこれをオーバーサンプリング座標系で処理し、寄与率 α をピクセルごとに求めてから α 合成を行うように拡張するとよい。ここから先は単なる力仕事である。また、`aa_lines()`関数はただ1本の線分ではなく数個~数百個の点列を結んで連続描画を行うので、それ相応の処理も考える必要がある。

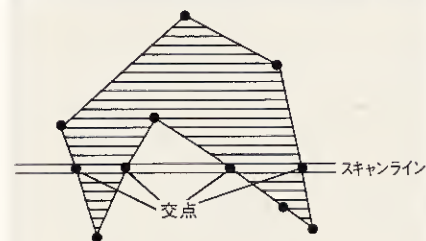
特にひとつの線分から次の線分に移るときは、前者の終点での寄与率を記憶しておいて後者の始点へとつなげていかなければならない。線分1本1本ごとに寄与率を初期化していたのでは、線分の継ぎ目継ぎ目でピクセルが暗くなってしまうからである(これは現実に失敗した)。

さきほどのめかしておいた欠点を説明しよう。`aa_lines()`関数では、傾きが小さいときはx方向に、傾きが大きいときはy方向に処理するようにループを組んでいる。また、太い線分といっても前述の通り平行四辺形で近似しているだけである。

そこで次のような事態は当然予想される。幅の太い曲線を描く場合を考えよう。その傾きは最初大きくてだんだん小さくなっていく。最初はy方向で処理していたのが、ある1点を境にx方向で処理するようになる。ここで曲線は、実にみつともないことに、まるでぼきんと折れたように欠けてしまうのだ。残念ながらこれを解消するうまい方法が考え出せなかった(下手な方法なら考えられないこともない)のでそのままにしてある。で、たいへん申し訳ないが、対抗策として、

- ・あまり太い線分は描かせない
- ・太い線分を描かせる場合は、傾きをうまくコントロールして曲線が折れないように工夫する
- ・どうしても自由な傾きで太い曲線を描きたいのであれば、面倒でも「太い曲線の輪

図4 ソリッドスキャンコンバージョン



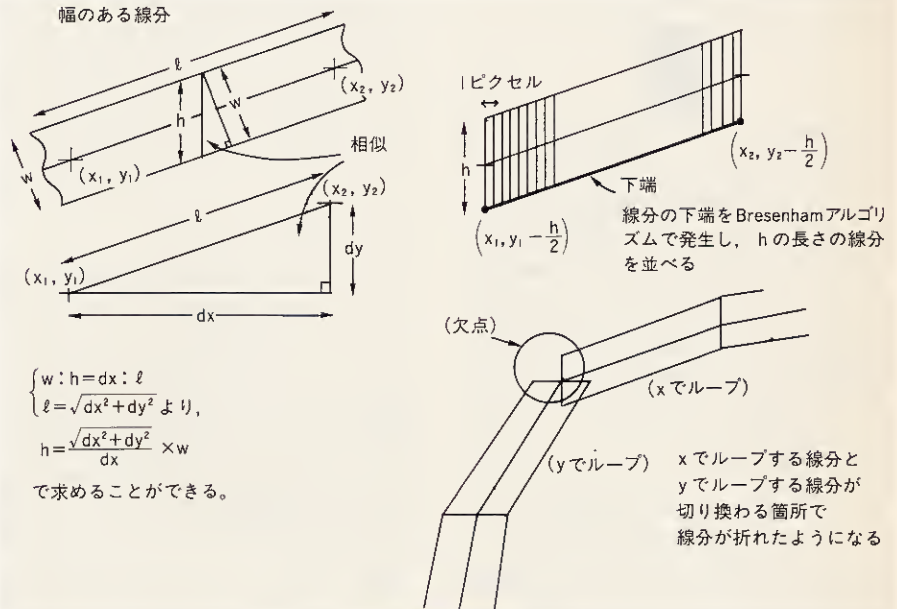
ソリッド領域をスキャンライン単位に細分する。スキャンラインと輪郭の交点は、Bresenhamアルゴリズムで求める。

郭」を作り、次のスキャンコンバージョン `aa_scanconv()` 関数で描かせる。スキャンコンバージョンのほうはどんな曲線に対しても破綻することはないなどとしていただきたい。

ソリッドスキャンコンバージョン

多角形を描画するもうひとつの方法で、上の`aa_lines()`がワイヤーフレームモデルだとしたら、こちらの`aa_scanconv()`はサーフェスモデルだといえるし、2次元ソリッドモデルだともいえる。要す

図3 線分の描画



リスト2

```
10 screen 1,3,1,1
20 fill( 0,0,47,47,rgb(0,0,31) )
30 symbol( 1,1,"色即",1,1,2,rgb(31,0,0),0 )
40 symbol( 1,25,"是空",1,1,2,rgb(31,0,0),0 )
50 symbol( 0,0,"色即",1,1,2,rgb(28,28,0),0 )
60 symbol( 0,24,"是空",1,1,2,rgb(28,28,0),0 )
70 tile_get( 0,0,0,47,47 )
80 tone_get( 0,0,0,47,47 )
90 fill( 0,0,47,47,rgb(16,16,16) )
100 tone_get( 1,0,0,47,47 )
110 wipe()
120 dim int p(10,2)={3,1,0
130     ,128,128,0
140     ,256,384,0
150     ,384,256,0}
160 pts_oversample( p )
170 dim int p1(10,2)={7,1,0
180     ,64,128,0
190     ,128,384,0
200     ,192,128,0
210     ,256,384,0
220     ,320,128,0
230     ,384,384,0
240     ,448,128,0}
241 dim int p2(2000,2)
242 pts_oversample( p1 )
243 pts_curve( p1,8,32,p2 )
244 whitepaper()
245 aa_lines( p2,0 )
250 /*aa_scanconv( p,1,0,0,0)
260 /*whitepaper(): aa_scanconv( p,0,rgb(31,0,0),1,0 )
270 aa_scanconv( p,1,0,1,1 )
```


寄ろう。まず多角形を細分する作業は、多角形とスキャンラインの交点の座標を求める処理に相当するが、これは輪郭をBresenhamアルゴリズムで発生すれば容易に求めることができる。またスクリーンに張り付ける作業は、求めた交点の間に線分を引く処理に相当する。やはり詳しい話は1989年7月号に譲る。

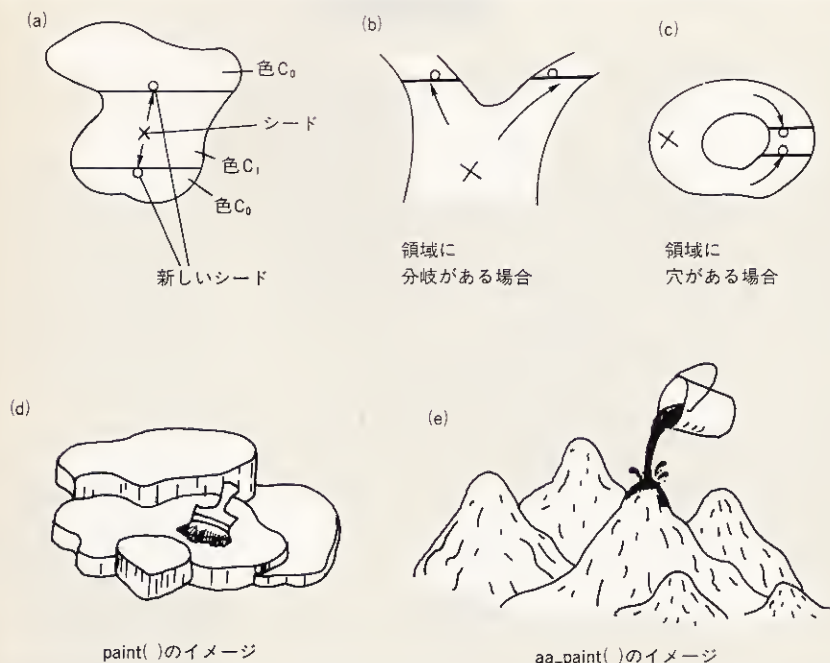
アンチエイリアシング化に際しては、先ほどのaa_lines()と同様のことをする。まずすべてオーバーサンプリング座標系で計算する。もちろんスキャンラインではなく、サブスキャンライン単位で処理するのである。そしてピクセルごとに寄与率 α を求めて背景と α 合成を行う。

ペイント

今回構成したペイントのアルゴリズムは実は本邦初ではないかと自負している（もちろん井の中の蛙かもしれない）。使い方の説明のところで、anti.fnc組み込みのaa_paint()と標準のpaint()の両方でペイントを実行してもらって、結果の違いを見ていただいた。paint()のほうでは、内側は赤いものの、曲線の緑の部分に白っぽいゴミが残り、変だった。対してaa_paint()だと隅々まできっちりと赤く塗ることができる。

ペイントのアルゴリズムをご存じだろうか。コンピュータグラフィックではシードフィルと呼ばれることもある。まず出発点

図5 ペイント



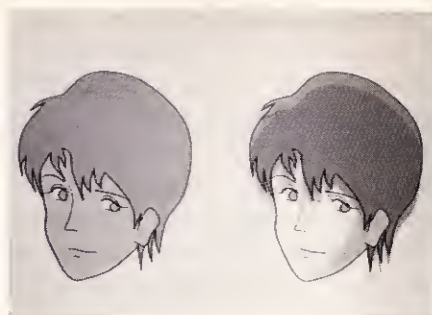
がある。これをシード（種）と呼ぶ。シードとなったピクセルの色c₀を記憶しておく。あとは、シードと同じ色をしていて、なおかつシードから到達可能なピクセルをすべてピックアップして、目的の色c₁で塗りつぶす。

「到達可能なピクセルを探す」アルゴリズムで一般的なのはFIFOバッファを使うアルゴリズムである。FIFOはファイフオと読み、先入れ先出し(First In First Out)方式でデータを格納する倉庫のようなものである。待ち行列といったり、キュー(queue)といったりする。ついでにスタックは後入れ先出し(Last In First Out)の倉庫で、LIFO（ライフオ）バッファといえる。

まずシードから左右を（同じスキャンライン内で）サーチしていく。サーチはピクセルの色がc₀以外の色になったところで止める。左右ともにサーチが終わったとき、そのスキャンラインの中で到達可能なピクセルがピックアップされたことになる。それを色c₁で塗りつぶす。

次に、上で作った到達可能な区間の上下のピクセルの色を調べる。もしその中に色がc₀のピクセルがあれば、その座標をFIFOバッファに入れる。色がc₀の領域は1本のスキャンラインにひとつとは限らない。途中でふたまたに分かれることもあるだろう。そうした領域の代表点を過不足なくバッファに入れるように、コーディングの際は工夫する。

ひとつのシードについての処理がひとと



タイトルも使える

おり終わったら、FIFOバッファから座標を1組取り出してきて、新しいシードにする。もしそのシードがすでに色c₁で塗りつぶしてあった場合は、そのシードを捨てる。そんなことが起こるのかと不思議に思う方もいらっしゃるだろうが、たとえばドーナツのように穴のあいた領域をペイントするときは、ぐるっと回ってきた色c₁の領域がぶつかるので、FIFOバッファに入れたときは色がc₀でも、FIFOバッファから出すときは色が変わっているということも起こりうるのだ。衝突したときに、どちらかのシードが無効になるわけだ。

さて、シードが有効なときは、そのシードから出発して上と同じことを繰り返す。そしてFIFOバッファが空になったとき、ペイントも終わる。

以上はふつうのペイントのアルゴリズム。だがオリジナルのペイントルーチンでも、基本は同じである。やはり到達可能なピクセルをピックアップし、シードを更新しつつ色を変えていけばよい。違うのは、「到達可能」を判定する条件である。ふつうのペイントでは、シードの色と同じであることがその条件であった。しかしここではアンチエイリアシングをかけた領域には対応できない。なぜなら、アンチエイリアシングは緑の色を少し暗くすることで滑らかに見せているものだから、当然緑の部分はシードと色が同じはずはないのである。したがって緑まできっちりと塗ることは不可能。

そこで新しい条件を作る必要がある。その条件とは極めて簡単で、現在いるピクセルと同じか、またはより暗いピクセルをたどっていくのである。こうすればサーチは暗いほうへ暗いほうへと進んでいき、明るくなりそうなところで止まる。これなら黒い線で囲んだ内側なら確実に隅まで塗ってくれるし、隣の白い部分にはみ出すこともない。

イメージとしては、山の頂上からペンキを流す図を想像していただきたい。ペンキは下へ下へと流れ、一番低いところで止ま

る。ほかの山を上っていくようなことはしない。

注意をうながしておくが、頂上、つまり白い部分の一番明るい点にシードを置かないと、やはり正確に塗ってはいくれない。

以上からもわかるとおり、オリジナルのペイントアルゴリズムでは、オーバーサンプリング座標は使わない。かわりに、ピクアップした点の輝度を寄与率 α のように考えて（というよりもアンチエイリアシング描画におけるピクセルの輝度はもともと α を反映したものなのだが）、 α 合成に似たことを行う。単純に色c1で塗るのではなく、

c1に α をかけた色で塗るのだ。

したがって、ペイントずみの領域を判定するには、色がc1であるかどうか、という見分け方が使えない。ペイントずみの領域には少し暗いc1というのもある。そこで今まで出番のなかった輝度ビットをフラグとして使うことにした。X68000のカラーコードは16ビットで、上から5ビットずつ緑、赤、青の3原色が割り当てられる。そして最下位の1ビットが輝度ビットなのである。RGBと独立になっているので妙な用途に使われることが多い。Z'sSTAFFでもマスキングに用いている。

CとBASICの相性

のコードを直接埋め込むという技が用意されている。まっとうなCコンパイラなら必ず使えるこの技は、当然X68000上のCコンパイラ、つまり標準のXCでも本誌6月号の付録ディスクで配布したGCCでも使える。ただ両コンパイラでのインラインアセンブラの使い方は少し違っていて、XCでは、

```
#asm
    lea __errmsg, a0
#endasm
であるが、GCCでは、
asm ( "lea __errmsg, a0" );
```

である。ここで、__errmsgはエラーメッセージを格納しているアドレスである。

GCCは本来の活動の舞台がUNIXなので、GCCの書き方はUNIX標準のCと同じである。C言語界全体を見渡せばむしろXCの作法がローカルな部類に入るのだろうが、そんなことはX68000でプログラムを作っている僕らにはなんの関係もない。どうにかして両者の違いを吸収する必要がある。

GCCのドライバはコンパイルに際して、プリプロセッサに、

```
#define __GCC__
と指定したのと同じことを自動的に行う。ま、環境変数みたいなものだ（本当は全然違う）。今回はこれを利用して条件コンパイル（#if～#else～#endif）をする方式を採用した。
```

```
#ifndef __GCC__
    asm ("lea __errmsg, a0");
#else
#asm
    lea __errmsg, a0
#endasm
#endif
```

しかし読者の方はこんな面倒なことをする必要はない。各自の使いたいコンパイラにあわせて部分だけを打ち込めばそれでよい。

お断りしておくまでもないと思うが、GCCだけ手に入れてもコンパイルはできない。コンパイラに際してはアセンブラとリンカとXCのライブラリが必要なので、XCつまりC compiler PR0-68Kは必ず持っていないとてならない。

ちなみにエラーメッセージであるが、グローバル変数の文字列として宣言するのがコツである。関数の外側で、文字列（char型配列）へのポインタとして、たとえば、

```
unsigned char errmsg[] = "エラーだよ";
と宣言するとよい。こちらの変数名の頭にはア
```

また、 α の値は、aa_paint()では赤成分の輝度を代表で持ってくることにした。これによりどんな不都合が起こるかという、たとえば真っ青な領域は塗れないのである。赤成分がないので、全部黒と見なされるのだ。その他、明るいところから暗いところへと塗るアルゴリズムのため、暗いところから出発して明るいほうに塗っていくような塗り方もできない。

以上のように妙な制限が多いので、白地に黒く線を描いてその中を塗るという使い方をすすめる。ついでにもうひとつについておくと、あまり細い領域を塗ろうとす

ンダーバー（　）がつかないことに注意。コンパイラは、ソース中のラベル（関数名や静的変数名）にアンダーバーをひとつつけてアセンブラに渡すが、すでに述べたとおり、インラインアセンブラの中身にはいっさい手を出さないの、こんな配慮が必要である。いくらかCで書けるといっても、アセンブラの知識が少しはないと、外部関数は書けないのだ。

さて順番が前後してしまったが、外部関数ヘッダである。これはもう純粋にアセンブラで書かないとしようがない。もちろんインラインアセンブラは使えるが、上述の条件コンパイルと同じことを2回書く必要がある、量が多いだけにユウウツである。ま、関数内で渡さなくてはならないエラーメッセージならともかく、ヘッダである。無理にCのソースリストの中に埋め込む必要もない。ヘッダは独立なファイルにした。それがantisである。

そしてもっともやっかいなのが引数の渡し方である。Cは引数を4バイトないしは8バイト単位でスタックに積み、関数に渡す。むろん呼ばれた関数側でも4バイト、8バイト単位で受ける。ところがX-BASICは引数を10バイト単位でスタックに積み、外部関数に渡す。10という数字はCにとってはとても半端な数字である。おかげでBASICからもらってきた引数を、Cのほうでストレートに受け取ることができなくなってしまっている。

で、これもしかたなくアセンブラで記述しなくてはならないのだろうかと思われた。ところがどっこい、Cの柔軟性をあなどってはいけない。引数リストを2バイト単位にばらせばどんな引数でも受けられるというのが鍵である、10は2で割り切れるのだから。

具体的には、まずダミー型を用意する。その名もずばり、DUMMY型（正体はただのintだが）。その引数dummyを指すポインタ&dummyを、2バイト整数の配列par[]へのポインタにキャストするのである。これでどんな引数が来ても大丈夫だ。引数のアクセスについてはマクロをしこたま使ったので、それほど関数本体では苦勞せずにすむだろう。しかし泥臭さには拭いがたいものがある。

引数リストの構造などは説明すると長くなるし、マクロの使い方さえ理解すれば十分だと思うのでもうこれ以上は説明しないが、もっと詳しく知りたい方は、本文の最後に掲げた参考文献をご覧ください。親切かつエレガントな技法に出会えるであろう。

X68000以前は、BASICの機能拡張といえば、メモリの空きエリアを捜して処理ルーチンを組み込んだり、パッチを当てたりといった、どことなく超絶技巧の香りが漂う技術であった。X-BASICでは、機能拡張を正式に許し、その仕様を公開している。さらに書くことと思えばCで書いたっていいのである。この姿勢には頭の下がる思いである。と同時にプログラマが甘やかされそうな気がしなくもない。

さて、そのX-BASICの外部関数はCで書くことができないのだが、いくつかの制限がある。

Cで素直に書けない部分について

いきなり矛盾したことをいっているようだが、BASICインタプリタと外部関数のインタフェースを取る段階で、どうしても純粋なCだけでは無理な部分があるのである（しかし素直でなくなりさえすれば簡単に書ける。ここがCの頼もしさであり、同時に怖さでもある）。具体的には、

- ・外部関数のヘッダ
 - ・引数を渡す
 - ・戻り値を返す
 - ・外部関数エラーのコードを返す
 - ・エラーメッセージのアドレスを返す
- 部分である。

このうち戻り値に関しては、今回作った関数はみんなvoid型ということにしてしまったので問題は起きない。

それにもかかわらず、Cのソースリストでの関数の戻り値がint型（typedefを使ってFUNC型としてはあるが）なのは、Cの関数の戻り値を実はBASIC側ではエラーコードとして受け取るためである。Cの関数は（整数型の）戻り値をd0レジスタに入れてリターンするというしきたり(?)があり、またBASICのエラーコードはd0で受け取るという規則になっている。return(0)で戻れば関数が無事に終わったことを、return(1)で戻ればなにかトラブルが起きたことをBASICに知らせることができる。エラーが起きたことがわかれば、インタプリタはエラーメッセージを出し、ビーブ音とともにプログラムの実行を中断してくれる。結局どちらもCで書くことができるのでこれも問題ない。

エラーメッセージが問題である。a0レジスタにエラーメッセージの先頭アドレスを入れて返さなくてはならない。これはさすがにCで書くことはできない。しかしC言語には、インラインアセンブラといって、ソース中にアセンブラ

ると、途中でペイントが止まってしまうことがあるが、これはふつうのペイントでも状況は同じであろう。このペイントはかなり好条件でないと働いてくれない、わがままペイントルーチンなのであった、残念ながら。

タイルとトーン

スキャンコンバージョンaa_scanconv()とペイントaa_paint()では、タイルとトーンを使うことができる。使い方はZ'sSTAFFのタイル&トーンとはほぼ同じで、描画の色にはカラーコード(単色)とタイルパターンのいずれかが選べ、トーンは使うか使わないかが選べる。

さらにスキャンコンバージョンでは、トーン指定の際に下地が透けて見えるか、それとも単に塗りつぶすのかを選ぶこともできる(ペイントは、そもそもアルゴリズム自体が下地の存在に大きく依存しているので、下地は透けて見えるのが当然なのである)。これにより、スクリーントーンを貼るのと同様の効果を狙っている。

ただし、下地が透けるモードでは、背景が黒いところにどんなトーンを張り付けてもなにも出ないので注意。スクリーント

ンもペイントと同様、白地に引いた黒い線で絵を描き、その上に貼るのが基本である。

タイルやトーンのパターンの登録の手順について。まず画面に基本パターンを描いておいて、それをtile_get()関数やtone_get()関数で取り込んで登録する。トーンは例によって、取り込んだパターンのうち、赤成分だけを見ている。まあモノトーンで描いておけば安心。また、通常のカラールコードでは明るい(白に近い)色のほうが値が大きいが、トーン登録に限っては、暗い(黒に近い)ほうがトーンの色が濃いとみなされる。これはZ'sSTAFFをまねたのだが、こちらのほうがわかりやすいようだ。

パターンを登録しておけば、ペイントでもスキャンコンバージョンでも、タイル番号やトーン番号を指定すれば呼び出すことができる。いろいろ指定して使い方を覚えていただきたい。

登録は本番の描画に先立ってやっておいたほうが、画面が乱れずにすむ。また一度登録しておけば、BASICを終わるまでパターンは消えないことになっている(BASICの変数とは別の場所に領域を確保している)。だから標準的なタイル/トーンパターンを定義するプログラムを描画プログラ

ムとは別に作り、BASIC起動時に一度だけrunしておけば、あとはそのパターンがずっと使える。

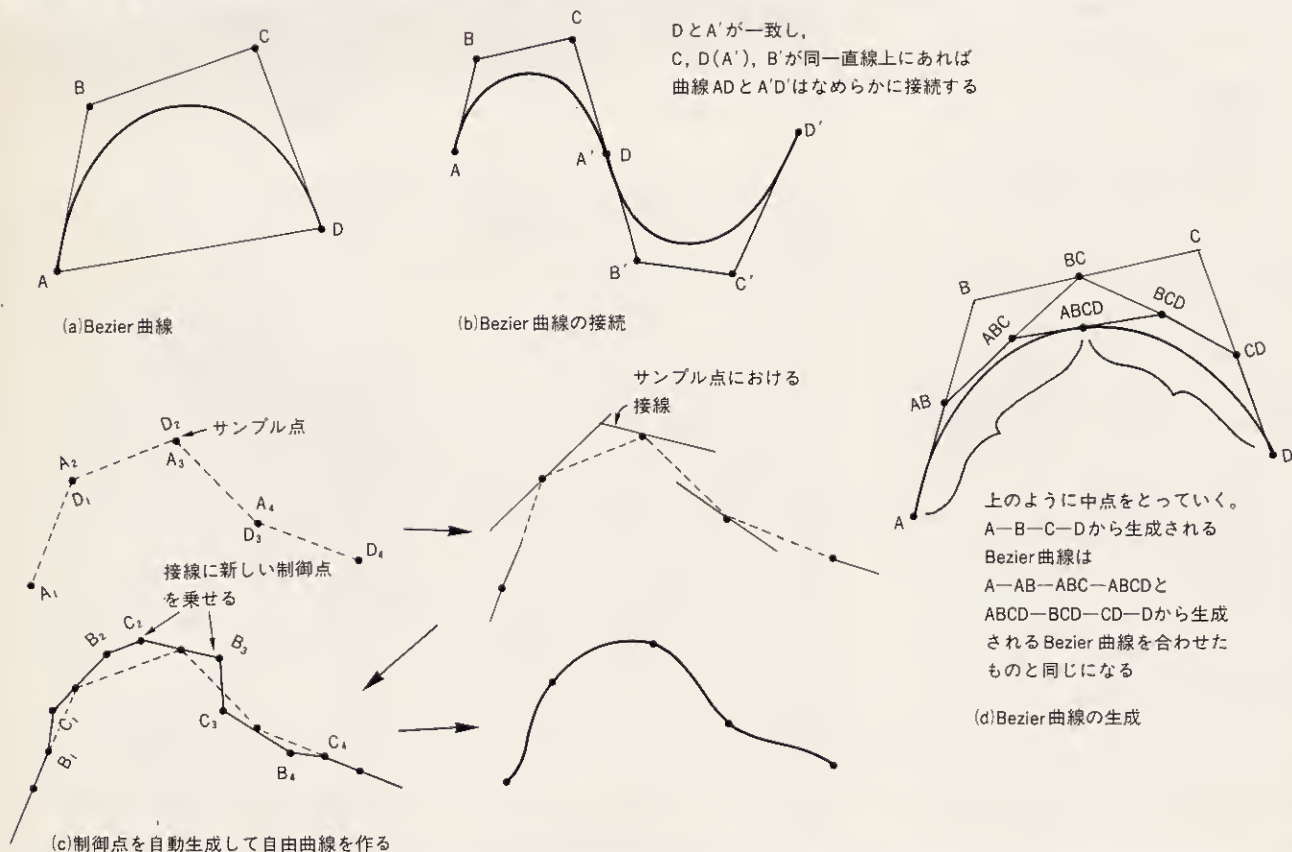
自由曲線

自由曲線にはBezier(ベジエ)曲線を使っている。Bezier曲線は4個ひと組の制御点を取る曲線である。その4つの制御点を順にA,B,C,Dとすると、Bezier曲線は次のような特徴を持つ。

- ・制御点Aから出発し、そこでは線分ABに接している。
- ・四角形ABCDの中に入る。
- ・制御点Dで終わり、そこでは線分CDに接している。

Bezier曲線は与えられた制御点すべてを通るわけではない(制御点BとCは通らない)が、これでは使いにくい。ふつうのユーザーなら、画面上にぽんぽんと点を置いて、その点を通る滑らかな曲線がほしい。かといっていちいち制御点BやCを手計算でつけ足していたのでは使いものにならない。下手な計算をすると、隣り合ったBezier曲線が滑らかにつながってくれないということになる。点列に記録されたすべての点を通る滑らかな曲線を生成するために

図6 Bezier曲線



は、制御点BやCをうまく自動的に計算して発生させる必要がある。

困っていたところで、以前に大学のコンピュータグラフィックの演習でうまいアルゴリズムを習ったことを思い出した。ここではそれを借用している。

その考え方を簡単に説明しておく。点列に記録されている点がサンプル点になる。まず各サンプル点上で、目的の曲線の接線を求める。次にその接線上に制御点BとCを乗せる。サンプル点は制御点AおよびDになる。こうしてできた制御点でBezier曲線を描かせると、曲線はサンプル点を通ってくれるし、しかもそのサンプル点上でなめらかにつながってくれる。

最後に、制御点A~Dが与えられたときのBezier曲線の発生のしかたを説明しよう。制御点の中点どうしをつないでいって新しい制御点を発生する。新しい制御点は2組できるのだが、それら発生する2つのBezier曲線をつなぎ合わせると、求めるBezier曲線が得られるのである。2分割して統合するのだから、再帰が使える。再帰的に新しい制御点を発生し、十分制御点の間隔が短くなったところで再帰を打ち切る。そんな制御点なら、いきなり線分でつない

でしまっても、十分滑らかな曲線に見える。そのレベルまで再帰を続けられればいいというわけ。

最後に

漫画家の道具はペンとインクと墨とスクリーンと、ほかになにがあるかはよく知らないが、その真似ごとを、ある程度のクオリティでできるようにはなったと思う。それでも動作テスト用のサンプルを作ろうとしてやっぱり嫌だなと思ったのは、どうしても数値を意識しておかないとなにも作れないので、いきおいつまらない図形で我慢してしまうところである。

このままでは人間デジタイザやドッターなみの忍耐力が必要だ。マウスでばつぱと描けるのが理想であろうが、それにはどうしてもマウスの動きに追従できるだけのレスポンスがいる。いっそアセンブラで全部書き下ろそうかと思ってしまいが、いまはコンパイラの力だけが頼りという状態だ。

ともあれ、アンチエイリアシングを手動ドット打ちなど使わずに実現できる可能性は示せたと思う。ジャギーフリーのグラフィックプリミティブを装備したペインティン

グツールというのはまだまだ先の話であらうが、その目標への第一歩としてこの外部関数をお使いいただければ幸いである。

参考文献

- (X-BASICの外部関数をCで書く方法について)
- ・C調言語講座PRO-68K第1回 まずはprintfより始めよ、祝 一平、Oh!X 1988年7月号, pp. 98-104
- ・C compiler PRO-68K プログラマーズマニュアル
- ・X 68000 BASIC入門 最終回 必殺サンプリング戦法、中森 章、Oh!X 1988年7月号, pp. 129-136
- ・Oh!X質問箱、村田 敏幸、Oh!X 1988年12月号, pp.129-167
- (幅のある線分について)
- ・アルゴリズムとプログラムによるコンピュータグラフィックス [I], S.Harrington著、郡山彬訳、マグロウヒル, pp.32-33
- (ソーティングアルゴリズムについて)
- ・PascalとCプログラムによるアルゴリズムとデータ構造ハンドブック、G.H.Gonnet著、玄光男・荒実・松本直文共訳、啓学出版, pp.129-136
- (Bezier曲線の制御点を自動生成することについて)
- ・コンピュータ・グラフィックスの基礎、鈴木賢次郎、長島忍、鈴木宏正, pp.A18-A20
- (Bezier曲線の再帰的分割による構成法について)
- ・アルゴリズムとプログラムによるコンピュータグラフィックス [II], S.Harrington著、郡山彬訳、マグロウヒル, pp.539-543

表1 関数リファレンス

オーバーサンプリング倍率はソースリスト (anti.h) のOVERSAMPLEの値を書き換えることで変えることができるが、今回は8倍オーバーサンプリングとした。

点列のフォーマット

int pts (n, 2) で宣言する。nは格納できる点列の長さの最大値。

(ヘッダ情報)

pts (0, 0) ...点列の長さ、頂点数 (≤n)

pts (0, 1) ...点列のタイプ (0のとき片道通行, 1のとき循環する)

pts (0, 2) ...オーバーサンプリング倍数 (点列がオーバーサンプリング座標のときには8が入る)

($1 \leq i \leq \text{pts}(0, 0)$ なる頂点iの情報)

pts (i, 0) ...x座標

pts (i, 1) ...y座標

pts (i, 2) ...幅 (オーバーサンプリング座標での値。これが8だと1ピクセル分の幅になる)

関数リファレンス

* どの関数にも、戻り値はない。

pts_oversample (pts)

(引数)

int pts (n, 2)

(機能)

通常のサンプリングレートで記述された点列 pts をオーバーサンプリング座標に変換する。

(注意)

点列がオーバーサンプリング座標かどうかは、pts (0, 2) の値で調べる。ここにオーバーサンプリング倍数 (8) が入っていれば、その点列はオーバーサンプリング座標である。

関数のうち、点列を引数にとるものは、オーバーサンプリング座標に変換しないと使えない。

pts_curve(pts1, w1, w2, pts2)

(引数)

int pts1(n1, 2), w1, w2, pts2(n2, 2)

(機能)

点列 pts1 の各頂点を通る自由曲線を生成し、点列 pts2 に格納する。その際、始点と終点での幅を w1, w2 とし、そのあいだの線の幅を線形補間する。

(注意)

曲線を微小線分で近似するので点列 pts2 は多少大きめに取る。

場合にもよるが、pts (1000, 2) 程度にしておけばよい。

配列の大きさが不足しているとエラーになる。

pts_append(pts1, pts2)

(引数)

int pts1(n1, 2), pts2(n2, 2)

(機能)

点列 pts1 と pts2 をつなげ、pts1 に格納する。

(注意)

pts1 の最後の点と pts2 の最初の点が一致するように pts2 を移動してからアペンドする。

pts1 の大きさ (n1の値) は、新しい点列の長さ、つまり(pts1(0, 0)+pts2(0, 0))以上用意しておくこと。

新しい点列のタイプ (片道通行か循環するか) は、もとの点列のうち pts1 のタイプにあわせる。

pts_move(pts1, x, y, pts2)

(引数)

int pts1(n1, 2), x, y, pts2(n2, 2)

(機能)

点列 pts1 をオフセットx,yで点列 pts2 に移動する。

(注意)

オフセットはオーバーサンプリング座標で指定すること (各座標を8倍する)。

オフセットをx,y共に0とした場合は、点列コピーの役割も果たす。

aa_lines(pts, c)

(引数)

int pts(n, 2), c

(機能)

点列 pts に沿って、線を色 c で連続描画する。線の幅は各頂点に記録されている値を用いる。

(注意)

線の幅が太すぎると、表示が一部欠けることがある。

aa_scanconv(pts, cmode, c または n_tile, tmode, n_tone)

(引数)

int pts(n, 2), cmode, c, n_tile, tmode, n_tone

(機能)

点列 pts を輪郭とする多角形の内部を塗りつぶす。

cmode=0...c で指定される色(単色)で塗る。

1...n_tile で指定されるタイルパターンで塗る。

tmode=0...トーンは用いない。下地の色と関係なく塗る。

1...n_tone で指定されるトーンを用いる。下地の色と関係なく塗る。

2...トーンは用いない。下地が透けて見える。

3...n_tone で指定されるトーンを用いる。下地が透けて見える。

(注意)

下地は、白地に黒い線を引いた場合を想定している。

下地のうち赤成分のみを取っている(青、緑成分は無視)ので、思いどおりの結果が出ないこともある。

tmode=2, 3 の場合は、黒い背景の場所に塗ってもなにも描画しない。

aa_paint(x, y, cmode, c または n_tile, tmode, n_tone)

(引数)

int x, y, cmode, c, n_tile, tmode, n_tone

(機能)

白地の中の点(x, y)を出発点として、黒い線で囲まれた閉領域を塗りつぶす。

aa_lines() で描画されたような、境界のはっきりしない領域も塗る。色(またはタイルパターン)、トーンは aa_scanconv() に準ずる。

(注意)

座標(x, y)には、オーバーサンプリングでないふつうの座標を指定すること。

境界判定アルゴリズムの関係上、あまり狭い部分を塗ることはできない。

白地から黒い境界に向かって塗るので、黒地から塗ることはできない。

輝度ビットをペイント済みフラグとして用いている。

tile_get(n, x1, y1, x2, y2)

(引数)

int n, x1, y1, x2, y2

(機能)

タイル番号nのタイルパターンをグラフィック画面の(x1, y1)-(x2, y2)の領域から取り込む。

(注意)

座標(x1, y1), (x2, y2)には、オーバーサンプリングでないふつうの座標

を指定すること。

取り込むパターンの大きさには制限がある (anti.h で定義されている T_SIZE の値)。

(x1, y1) が始点で (x2, y2) が終点。大小関係を変えれば、反転したパターンも取り込む。

tone_get(n, x1, y1, x2, y2)

(引数)

int n, x1, y1, x2, y2

(機能)

トーン番号nのトーンをグラフィック画面の(x1, y1)-(x2, y2)の領域から取り込む。

(注意)

座標(x1, y1), (x2, y2)には、オーバーサンプリングでないふつうの座標を指定すること。

取り込むパターンの大きさには制限がある (anti.h で定義されている T_SIZE の値)。

(x1, y1) が始点で (x2, y2) が終点。大小関係を変えれば、反転したパターンも取り込む。

トーンの濃さは赤成分から取る。黒に近いほど(輝度が低いほど)濃くなる。

whitepaper()

(引数)

なし

(機能)

白紙を作る

(注意)

fill (0, 0, 511, 511, 65534) と同じ。

輝度ビットが立っていないのに注意 (aa_paint() に支障をきたさないように)。

reverse()

(引数)

なし

(機能)

画面を反転させる

(注意)

どうしても黒地を aa_paint() で塗りつぶしたいときに使う。

いったん反転させて(白地になっている)塗り、もう一度反転させて戻す。

反転に際しては輝度ビットをいじらない。

maskclear()

(引数)

なし

(機能)

輝度ビットをすべて 0 にする。

(注意)

aa_paint() は、輝度ビットの立っているところでは使えない。

先にこのマスクをクリアしておくための関数。

リスト3

```
===== aa_scanconv.c =====
1: /***** アンチエイリアシング付きソリッドスキャンコンバージョン *****/
2:
3: #include <graph.h>
4: #include "anti.h"
5:
6: /* 多角形の辺... 線分の集合 */
7: typedef struct {
8:   int x, y, dx2, dy2, sx, ex, ry;
9: } EDGE;
10:
11: /* リスト処理用の定数 */
12: #define FIRST 0 /* アクティブエッジリストの最初の要素 */
13: #define LAST 1 /* アクティブエッジリストの最後の要素 */
14: #define FORWARD 0 /* アクティブエッジリストの次要素 */
15: #define BACKWARD 1 /* アクティブエッジリストの前要素 */
16: #define NEXT 1 /* フリーリストの次要素 */
17: #define NIL -1 /* リストの終端を示す */
18:
19: #define MAXACTIVE 128 /* 1本のスキャンラインを切れる辺の数 */
20: #define MAXEDGE 1024 /* 一度に処理できる辺の数 */
21:
22: EDGE edge[ MAXEDGE ];
23: EDGE *edgeptr[ MAXEDGE ]; /* activeedgeptr[ MAXACTIVE ];
24: int activelist[ MAXACTIVE ]; /* 2; /* アクティブエッジリストは双方向リスト・フリーリストと共用 */
25: int scanlinebuffer[ MAXACTIVE ]; /* 輪郭とスキャンラインの交点の x 座標 */
26:
27: /* タイルおよびトーン (ペイントと共有) */
28: extern unsigned char Color[3]; /* R,G,B の 3色 */
29: extern unsigned char Tile[ N_TILE ][ T_SIZE ][ T_SIZE ][3]; /* R,G,B の 3色 */
30: extern unsigned int Tile_x[ N_TILE ]; /* タイル x のタイル番号 */
31: extern unsigned char Tone[ N_TONE ][ T_SIZE ][ T_SIZE ]; /* 単色 */
```

```
32: extern unsigned int Tone_x[ N_TILE ], Tone_y[ N_TILE ]; /* トーンの大きさ */
33:
34: extern unsigned int Alpha[ N_PIXEL ]; /* 1 スキャンラインおんの透過率バッファ */
35: extern unsigned short Sibuf[ N_PIXEL ]; /* 1 スキャンラインおんのフレームバッファ */
36:
37: extern unsigned char OVERSAMPLE_NOTYET[];
38:
39: extern int tile_tone_check(); /* タイル・トーンの指定が正しいかどうか調べる */
40:
41: unsigned char TOO_COMPLEX[] = "入力した点列が複雑すぎます";
42:
43: FUNC aa_scanconv( dummy )
44: DUMMY dummy;
45: /* PTS pts, int cmode, c/n_tile, tmode, n_tone */
46: {
47:   PTS pts;
48:   int cmode, c, n_tile, tmode, n_tone;
49:   int tile_x, tile_y, tone_x, tone_y;
50:
51:   int n_edge, y_min, y_max;
52:   int n, i, j, d, fw, bk, x, x1, x2, y, y1, y2, tmp;
53:   int active[2], inactive, free;
54:   EDGE *tapp;
55:   unsigned int r, g, b, ri, gi, bi, v, vm, a, s;
56:
57:   ARGSET( dummy );
58:   ARGSET(1);
59:   pts=PARTYTOP(1);
60:   cmode=IVALUE(2);
61:   c=n_tile=IVALUE(3);
62:   tmode=IVALUE(4);
63:   n_tone=IVALUE(5);
```



```

64:
65: /* 点列のタイプ、つまり pts[0][1] は無視される ( CYCLIC として扱われる ) */
66: npts[0][0];
67: if ( n>MAXEDGE ) {
68: #ifdef _GNU_
69: asm ( " lea.l _TOO_COMPLEX,a1" );
70: #else
71: #asm
72: lea.l _TOO_COMPLEX,a1
73: #endasm
74: #endif
75: return ( 1 );
76: }
77: if ( pts[0][2]>OVERSAMPLE ) {
78: #ifdef _GNU_
79: asm ( " lea.l _OVERSAMPLE_NOTYET,a1" );
80: #else
81: #asm
82: lea.l _OVERSAMPLE_NOTYET,a1
83: #endasm
84: #endif
85: return ( 1 );
86: }
87:
88: if ( cmode==COLOR ) {
89: bl=BLUE( c );
90: rl=RED( c );
91: gl=GREEN( c );
92: }
93: i=tile_tone_check( cmode, n_tile, &tile_x, &tile_y, tmode, n_tone, &tone_x, &tone_y );
94: if ( i!=1 ) return ( 1 );
95:
96: y_min=N_PIXEL*OVERSAMPLE*2; y_max=-N_PIXEL*OVERSAMPLE*2;
97: for ( i=1; n_edge=0; i<n; i++ ) {
98: x1 = pts[i][0];
99: y1 = pts[i][1];
100: if ( i!=n ) { /* 通常の点 */
101: x2 = pts[i+1][0];
102: y2 = pts[i+1][1];
103: } else { /* 終点は始点とつながる */
104: x2 = pts[1][0];
105: y2 = pts[1][1];
106: }
107: if ( y2==y1 ) continue; /* 水平なエッジは使わない */
108: if ( y2<y1 ) { /* スキャンコンバージョンは上から下へ処理する */
109: tmp=x1; x1=x2; x2=tmp;
110: tmp=y1; y1=y2; y2=tmp;
111: }
112: if ( y_min>y1 ) y_min=y1; /* 図形の存在範囲を求める */
113: if ( y_max<y2 ) y_max=y2;
114: edge[n_edge].x = x1; /* Bresenham アルゴリズムの初期値 */
115: edge[n_edge].y = y1;
116: edge[n_edge].dx2= 2*ABS( x2-x1 );
117: edge[n_edge].ax= SGN( x2-x1 );
118: edge[n_edge].dy2= 2*( y2-y1 );
119: edge[n_edge].ry= ( y2-y1 );
120: edge[n_edge].e = -( y2-y1 );
121: edgeptr[n_edge]=(&edge[n_edge]);
122: n_edge++;
123: }
124: for ( d=n_edge; d>1; ) { /* エッジを始点が上にあるものから順にソートする */
125: if ( d<5 ) { /* Shell ソートを用いている */
126: d=1;
127: } else {
128: d=(5*d-1)/11;
129: }
130: for ( i=n_edge-1-d; i>0; i-- ) {
131: tmp=edgeptr[i]; /* ポインタだけを並べ替えてスピードアップを図る */
132: for ( j=i+d; j<(n_edge-1) && ((tmp->y)>(edgeptr[j]->y)); j+=d ) {
133: edgeptr[j]=tmp;
134: }
135: edgeptr[j-d] = tmp;
136: }
137: }
138: free=0; /* フリーリストの先頭の要素 */
139: active[FIRST]=active[LAST]=NIL; /* アクティブエッジリストを空にする */
140: inactive=0; /* アクティブでないエッジの先頭 */
141: for ( i=0; i<MAXACTIVE-1; i++ ) { /* フリーリストを初期化する */
142: activelist[i][NEXT]=i+1;
143: }
144: activelist[MAXACTIVE-1][NEXT]=NIL; /* フリーリストの終端 */
145:
146: for ( y=y_min; y<y_max; y++ ) {
147: /* 最初、またはスキャンラインの始めごとに、フレーム/バッファから取り込む */
148: if ( y==y_min || (y%OVERSAMPLE)==0 ) {
149: get( 0, PIX(y), N_PIXEL-1, PIX(y), Sbuf, N_PIXEL*sizeof(short) );
150: for ( x=0; x<N_PIXEL; x++ ) Alpha[x]=0;
151: }
152: /* 新しい active edge を探す (始点が現在のスキャンラインと重なるエッジ) */
153: while ( inactive<n_edge ) {
154: if ( edgeptr[inactive]->y == y ) break; /* ソート済みので打ち切り条件は素 */
155: if ( free==NIL ) { /* アクティブエッジリストの容量を超えた */
156: #ifdef _GNU_
157: asm ( " lea.l _TOO_COMPLEX,a1" );
158: #else
159: #asm
160: lea.l _TOO_COMPLEX,a1
161: #endasm
162: #endif
163: return ( 1 );
164: }
165: /* 新しいエッジをフリーリストから取ってくる */
166: if ( active[FIRST]==NIL ) { /* アクティブエッジリストが空の場合、新規に作る */
167: active[FIRST]=active[LAST]=free;
168: free=activelist[free][NEXT];
169: } else {
170: activelist[active[LAST]][FORWARD]=active[FIRST];
171: activelist[active[FIRST]][BACKWARD]=active[LAST];
172: }
173: free=free;
174: free=activelist[free][NEXT];
175: activelist[active[LAST]][FORWARD]=free;
176: activelist[free][FORWARD]=active[LAST];
177: activelist[free][BACKWARD]=active[LAST];
178: active[LAST]=free;
179: }
180: /* アクティブエッジとスキャンラインとの交点を求める */
181: n=0;
182: for ( i=active[FIRST]; i!=NIL; i=activelist[i][FORWARD] ) {
183: tmp=edgeptr[i];
184: if ( -(tmp->y)>0 ) { /* 寿命の切れたエッジはリストから削除する */
185: f=activelist[i][FORWARD]; /* 双方リストは削除が楽 */
186: bk=activelist[i][BACKWARD];
187: activelist[i][NEXT]=free; /* 削除したエッジはフリーリストにつなぐ */
188: free=i;
189: } else if ( f==NIL && bk==NIL ) {
190: active[FIRST]=active[LAST]=NIL;
191: continue;
192: }
193: if ( f==NIL ) {

```

```

194: active[LAST]=bk;
195: activelist[bk][FORWARD]=NIL;
196: continue;
197: }
198: if ( bk==NIL ) {
199: active[FIRST]=f;
200: activelist[f][BACKWARD]=NIL;
201: continue;
202: }
203: activelist[bk][FORWARD]=f;
204: activelist[f][BACKWARD]=bk;
205: continue;
206: }
207: scanlinebuffer[n++] = tmp->x; /* 交点をすべてバッファに入れる */
208: (tmp->y) += (tmp->dx2); /* Bresenham アルゴリズムで交点を求める */
209: while ( (tmp->y)>0 ) {
210: (tmp->x) += (tmp->ax);
211: (tmp->y) -= (tmp->dy2);
212: }
213: }
214: if ( n==0 ) continue; /* 交点がない */
215: scanlinebuffer[n]=N_PIXEL*OVERSAMPLE*2; /* 番兵 (sentinel) */
216: for ( i=n-2; i>0; i-- ) { /* 交点をソート (番兵つき挿入ソート) */
217: tmp=scanlinebuffer[i];
218: for ( j=i+1; tmp<scanlinebuffer[j]; j++ ) {
219: scanlinebuffer[j-1]=scanlinebuffer[j];
220: }
221: scanlinebuffer[j-1]=tmp;
222: }
223: if ( y<0 ) continue; /* y でクリッピングする */
224: if ( y>(N_PIXEL*OVERSAMPLE) ) break;
225: /* 奇数率αを更新する */
226: for ( i=0; i<n-1; i+=2 ) {
227: x1=scanlinebuffer[i]; x2=scanlinebuffer[i+1];
228: if ( x2<0 ) continue; /* x でクリッピングする */
229: if ( x1>(N_PIXEL*OVERSAMPLE) ) break;
230: if ( x1<0 ) x1=0;
231: if ( x2>(N_PIXEL*OVERSAMPLE) ) x2=(N_PIXEL*OVERSAMPLE-1);
232: if ( PIX(x1) == PIX(x2) ) { /* 2つの交点が同じピクセル内にある */
233: Alpha[PIX(x1)] += (x2-x1);
234: } else { /* 2つの交点が違うピクセルにある */
235: Alpha[PIX(x1)] += (OVERSAMPLE-SUBPIX(x1));
236: for ( x=PIX(x1)+1; x<PIX(x2); x++ ) {
237: Alpha[x] += OVERSAMPLE;
238: }
239: Alpha[PIX(x2)] += (SUBPIX(x2)+1);
240: }
241: }
242: /* スキャンラインの終わりに、または図形の最後、α 合成して出力 */
243: if ( y==(y_max-1) || SUBPIX(y)==(OVERSAMPLE-1) ) {
244: vm=OVER2; /* α の最大値 */
245: if ( tmode&ON ) { /* トーンあり */
246: vm=1MAX;
247: }
248: if ( tmode&TP ) { /* 下地が透けるモード */
249: vm=1MAX;
250: }
251: y1=PIX(y);
252: for ( x1=0; x1<N_PIXEL; x1++ ) {
253: a=Alpha[x1];
254: if ( a==0 ) continue;
255: if ( a>OVER2 ) a=OVER2;
256: if ( cmode==TILE ) { /* タイル/カーン */
257: b1=tile[n_tile][y1*tile_y][x1*tile_x][0];
258: r1=tile[n_tile][y1*tile_y][x1*tile_x][1];
259: g1=tile[n_tile][y1*tile_y][x1*tile_x][2];
260: }
261: s=Sluf[x1];
262: v=a;
263: if ( tmode&ON ) { /* トーンあり */
264: v+=Tone[n_tone][y1*tone_y][x1*tone_x];
265: }
266: if ( tmode&TP ) { /* 下地が透けるモード */
267: v=VALUE(s);
268: }
269: b=( BLUE(s) *(vm-v)+b1+v )/vm;
270: r=( RED(s) *(vm-v)+r1+v )/vm;
271: g=( GREEN(s) *(vm-v)+g1+v )/vm;
272: Sbuf[x1]=RGB( r, g, b );
273: }
274: put( 0, y1, N_PIXEL-1, y1, Sbuf, N_PIXEL*sizeof(short) );
275: }
276: }
277: return ( 0 );
278: }
279:
280: FUNC scanconv dummy; /* アンチエイシングなしのバージョン */
281: DUMMY dummy; /* 速いののでちょっとした確認には使える */
282: /* PTS *pts, int c */
283: {
284: PTS *pts;
285: int c;
286:
287: int n_edge, y_min, y_max;
288: int n, i, j, d, f, b, x1, x2, y, y1, y2, tmp;
289: int active[2], inactive, free;
290: EDGE *tmpp;
291:
292: ARGSET( dummy );
293: ARYSET(1);
294: pts=PARTOP(1);
295: c=IVALUE(2);
296:
297: n=pts[0][0];
298: if ( n>MAXEDGE ) {
299: #ifdef _GNU_
300: asm ( " lea.l _TOO_COMPLEX,a1" );
301: #else
302: #asm
303: lea.l _TOO_COMPLEX,a1
304: #endasm
305: #endif
306: return ( 1 );
307: }
308: if ( pts[0][2]>OVERSAMPLE ) {
309: #ifdef _GNU_
310: asm ( " lea.l _OVERSAMPLE_NOTYET,a1" );
311: #else
312: #asm
313: lea.l _OVERSAMPLE_NOTYET,a1
314: #endasm
315: #endif
316: return ( 1 );
317: }
318: y_min=N_PIXEL*2; y_max=-N_PIXEL*2;
319: for ( i=1; n_edge=0; i<n; i++ ) {
320: x1 = PIX(pts[i][0]);
321: y1 = PIX(pts[i][1]);
322: if ( i!=n ) {
323: x2 = PIX(pts[i+1][0]);

```



```

324: y2 = PIX(pts[i+1][1]);
325: } else {
326: x2 = PIX(pts[i][0]);
327: y2 = PIX(pts[i][1]);
328: }
329: if ( y2==y1 ) continue;
330: if ( y2<y1 ) {
331: tmp=x1; x1=x2; x2=tmp;
332: tmp=y1; y1=y2; y2=tmp;
333: }
334: if ( y_min>y1 ) y_min=y1;
335: if ( y_max<y2 ) y_max=y2;
336: edge[n_edge].x = x1;
337: edge[n_edge].y = y1;
338: edge[n_edge].dx2= 2*ABS( x2-x1 );
339: edge[n_edge].sx= SIGN( x2-x1 );
340: edge[n_edge].dy2= 2*( y2-y1 );
341: edge[n_edge].ry= ( y2-y1 );
342: edge[n_edge].e = -( y2-y1 );
343: edgeptr[n_edge]=( &edge[n_edge] );
344: n_edge++;
345: }
346: for ( d=n_edge; d<1; ) {
347: if ( d<5 ) {
348: d=1;
349: } else {
350: d=(5*d-1)/11;
351: }
352: for ( i=n_edge-1; i>=0; i-- ) {
353: tmp=edgeptr[i];
354: for ( j=i+d; j<n_edge-1; j+=d ) {
355: edgeptr[j-d] = edgeptr[j];
356: }
357: edgeptr[j-d] = tmp;
358: }
359: }
360: free=0;
361: active[FIRST]=active[LAST]=NIL;
362: inactive=0;
363: for ( i=0; i<MAXACTIVE-1; i++ ) {
364: activelist[i][NEXT]=i+1;
365: }
366: activelist[MAXACTIVE-1][NEXT]=NIL;
367: for ( y=y_min; y<y_max; y++ ) {
368: while ( inactive<n_edge ) {
369: if ( edgeptr[inactive]->y != y ) break;
370: if ( free==NIL ) {
371: if ( free==NIL ) {
372: #ifdef _GNUC_
373: asm ( " len.l _TOO_COMPLEX,a1" );
374: #else
375: #asm
376: len.l _TOO_COMPLEX,a1
377: #endasm
378: #endif
379: return ( 1 );
380: }
381: if ( active[FIRST]==NIL ) {
382: active[FIRST]=active[LAST]=free;
383: free=activelist[free][NEXT];
384: activelist[active[LAST]][FORWARD]=activelist[active[FIRST]][BACKWARD]=NIL;
385: } else {
386: f=free;
387: free=activelist[free][NEXT];
388: activelist[active[LAST]][FORWARD]=f;

```

```

389: activelist[f][FORWARD]=NIL;
390: activelist[f][BACKWARD]=active[LAST];
391: active[LAST]=f;
392: }
393: activeedgeptr[active[LAST]]=edgeptr[active[LAST]];
394: inactive++;
395: }
396: n=0;
397: for ( i=active[FIRST]; i!=NIL; i=activelist[i][FORWARD] ) {
398: tmp=activeedgeptr[i];
399: if ( --(tmp->y)<0 ) {
400: [active[i]][FORWARD]=0;
401: b=activelist[i][BACKWARD];
402: activelist[i][NEXT]=free;
403: free=i;
404: if ( f==NIL && b==NIL ) {
405: active[FIRST]=active[LAST]=NIL;
406: continue;
407: }
408: if ( f==NIL ) {
409: active[LAST]=b;
410: activelist[b][FORWARD]=NIL;
411: continue;
412: }
413: if ( b==NIL ) {
414: active[FIRST]=f;
415: activelist[f][BACKWARD]=NIL;
416: continue;
417: }
418: activelist[b][FORWARD]=f;
419: activelist[f][BACKWARD]=b;
420: continue;
421: }
422: scanlinebuffer[n++] = tmp->x;
423: (tmp->e) += (tmp->dx2);
424: while ( (tmp->e) >= 0 ) {
425: (tmp->x) += (tmp->sx);
426: (tmp->e) -= (tmp->dy2);
427: }
428: }
429: if ( n==0 ) continue;
430: scanlinebuffer[n]=N_PIXEL*2;
431: for ( i=n-2; i>=0; i-- ) {
432: tmp=scanlinebuffer[i];
433: for ( j=i+1; j<n; j++ ) {
434: scanlinebuffer[j-1]=scanlinebuffer[j];
435: }
436: scanlinebuffer[j-1]=tmp;
437: }
438: for ( i=0; i<n-1; i+=2 ) {
439: line( scanlinebuffer[i], y, scanlinebuffer[i+1], y, c, 0xFFFF );
440: }
441: }
442: return ( 0 );
443: }

```

リスト4

```

----- aa_lines.c -----
1: /* **** アンチエイリアシング付き点列描画 **** */
2: #include <graph.h>
3: #include <math.h>
4: #include "anti.h"
5: extern unsigned char Color[3]; /* 描画色 */
6: extern unsigned int Alpha[N_PIXEL]; /* 1ライン分の書き込みバッファ */
7: extern unsigned short Sbuf[N_PIXEL]; /* 1ライン分のフレームバッファ */
8: extern unsigned char OVERSAMPLE_NOTET[1];
9: int Direction; /* 描画中の点列の傾きが大きい小さいか */
10: void aa_line( x1, y1, x2, y2, w ) /* 1区間だけ描画(1本)描く */
11: int x1, y1, x2, y2, w;
12: {
13: double h0;
14: int newdirection, i, j, x, y, sx, dx, dx2, sy, dy, dy2, e, h;
15: int xa, xb, ya, yb;
16: unsigned int r, g, b;
17: unsigned int a;
18: unsigned short c, s;
19: x=x1; y=y1; /* Bresenhamアルゴリズムの初期設定 */
20: dx=ABS( x2-x1 );
21: sx=SIGN( x2-x1 );
22: dx2=dx*2;
23: dy=ABS( y2-y1 );
24: sy=SIGN( y2-y1 );
25: dy2=dy*2;
26: if ( dx==0 && dy==0 ) return;
27: e=0;
28: c=RGB( Color[1], Color[2], Color[0] );
29: if ( dx>0 ) { /* 傾きが小さい場合は x でループ */
30: if ( sx>0 ) {
31: newdirection=1;
32: } else {
33: newdirection=3;
34: }
35: }
36: } else { /* 傾きが大きい場合は y でループ */
37: if ( sy>0 ) {
38: newdirection=2;
39: } else {
40: newdirection=4;
41: }
42: }
43: if ( newdirection==1 || newdirection==3 ) { /* 傾きが小さい場合は x でループ */
44: /* Direction が切り換わる時、スクリーンから1ライン取り込む */
45: if ( Direction != newdirection ) { /* Direction=0 (最初) の場合も含む */
46: get( PIX(x), 0, PIX(x), N_PIXEL-1, Sbuf, N_PIXEL*sizeof(short) );
47: for ( i=0; i<N_PIXEL; i++ ) Alpha[i]=0; /* 書き込みバッファをクリアする */
48: Direction=newdirection;
49: }
50: /* 傾きから高さを計算する */
51: h0 = ( (double)w * sqrt( (double)dx*(double)dx+(double)dy*(double)dy ) / dx );
52: h = (int)h0;
53: y = -(int)(h0/2.0); /* 下端線の始点 */
54: for ( i=0; i<dx; i++ ) { /* x のループ */
55: /* 1ライン処理するときスクリーンから新しいラインを取り込む */
56: /* 始点ではラインを取り込むとは限らない (前の傾きの情報を読む) */
57: if ( (sx==1 && SUBPIX(x)==0) || (sx==1 && SUBPIX(x)==(OVERSAMPLE-1)) ) {
58: get( PIX(x), 0, PIX(x), N_PIXEL-1, Sbuf, N_PIXEL*sizeof(short) );

```

```

59: for ( j=0; j<N_PIXEL; j++ ) Alpha[j]=0; /* 書き込みバッファをクリアする */
60: }
61: ya=y; yb=y+h-1; /* 書き込み範囲を設定する */
62: if ( yb>0 && ya<N_PIXEL*OVERSAMPLE ) {
63: if ( ya<0 ) ya=0;
64: if ( yb>N_PIXEL*OVERSAMPLE ) yb=N_PIXEL*OVERSAMPLE-1;
65: if ( PIX(ya) == PIX(yb) ) {
66: Alpha[PIX(ya)] += (yb-ya);
67: } else {
68: Alpha[PIX(ya)] += (OVERSAMPLE-SUBPIX(ya));
69: for ( j=(PIX(ya)+1); j<PIX(yb); j++ ) {
70: Alpha[j] += OVERSAMPLE;
71: }
72: Alpha[PIX(yb)] += (SUBPIX(yb)+1);
73: }
74: }
75: e+=dy2; /* Bresenham アルゴリズムで下端線分を生成する */
76: if ( e>dx ) {
77: e=sy;
78: e-=dx2;
79: }
80: /* 1ラインごと (または終点) でα合成と出力を行う */
81: if ( i==dx-1 || (sx==1 && SUBPIX(x)==0) || (sx==1 && SUBPIX(x)==(OVERSAMPLE-1)) ) {
82: for ( j=0; j<N_PIXEL; j++ ) {
83: if ( (a=Alpha[j])>0 ) continue;
84: if ( a>OVER2 ) { /* 書き込み範囲をオーバー */
85: Sbuf[j] = c;
86: } else { /* そうでないならα合成 */
87: a=Sbuf[j];
88: b=( (OVER2-a)*BLUE(s) + a*Color[0] ) / OVER2;
89: r=( (OVER2-a)*RED(s) + a*Color[1] ) / OVER2;
90: g=( (OVER2-a)*GREEN(s) + a*Color[2] ) / OVER2;
91: Sbuf[j] = RGB( r, g, b );
92: }
93: }
94: put( PIX(x), 0, PIX(x), N_PIXEL-1, Sbuf, N_PIXEL*sizeof(short) );
95: }
96: }
97: } else { /* 傾きが大きい場合は y でループ、以下同様 */
98: if ( Direction != newdirection ) {
99: get( 0, PIX(y), N_PIXEL-1, PIX(y), Sbuf, N_PIXEL*sizeof(short) );
100: for ( i=0; i<N_PIXEL; i++ ) Alpha[i]=0;
101: Direction=newdirection;
102: }
103: h0 = ( (double)w * sqrt( (double)dx*(double)dx+(double)dy*(double)dy ) / dy );
104: h = (int)h0;
105: x = (int)(h0/2.0);
106: for ( i=0; i<dy; i++ ) { /* y のループ */
107: if ( (sy==1 && SUBPIX(y)==0) || (sy==1 && SUBPIX(y)==(OVERSAMPLE-1)) ) {
108: get( 0, PIX(y), N_PIXEL-1, PIX(y), Sbuf, N_PIXEL*sizeof(short) );
109: for ( j=0; j<N_PIXEL; j++ ) Alpha[j]=0;
110: }
111: xa=x; xb=x+h-1;
112: if ( xb>0 && xa<N_PIXEL*OVERSAMPLE ) {
113: if ( xa<0 ) xa=0;
114: if ( xb>N_PIXEL*OVERSAMPLE ) xb=N_PIXEL*OVERSAMPLE-1;
115: if ( PIX(xa) == PIX(xb) ) {
116: Alpha[PIX(xa)] += (xb-xa);
117: } else {

```



```

118: Alpha[PIX(xa)] += (OVERSAMPLE-SUBPIX(xa));
119: for ( j=(PIX(xa)+1); j<PIX(xb); j++ ) {
120:   Alpha[j] += OVERSAMPLE;
121: }
122: Alpha[PIX(xb)] += (SUBPIX(xb)+1);
123: }
124: }
125: e+=dx2;
126: if ( e>dy ) {
127:   x+=ex;
128:   e-=dy2;
129: }
130: if ( i==dy-1 || (sy==1 && SUBPIX(y)==0) || (sy==1 && SUBPIX(y)==(OVERSAMPLE-1)) ) {
131:   for ( j=0; j<N_PIXEL; j++ ) {
132:     if ( (a=Alpha[j])>0 ) continue;
133:     if ( a>OVER2 ) {
134:       Slbuf[j]=c;
135:     } else {
136:       a=Slbuf[j];
137:       b=( OVER2-a)*BLUE(s) + a*Color[0] /OVER2;
138:       r=( OVER2-a)*RED(s) + a*Color[1] /OVER2;
139:       g=( OVER2-a)*GREEN(s) + a*Color[2] /OVER2;
140:       Slbuf[j] = RGB( r, g, b );
141:     }
142:   }
143:   put( 0, PIX(y), N_PIXEL-1, PIX(y), Slbuf, N_PIXEL*sizeof(short) );
144: }
145: }
146: }
147: return;
148: }
149: FUNC aa_lines( dummy ) /* 関数本体、全点を描画する */
150: DUMMY dummy;
151: /* PTS *pts, int c */
152: {
153:   PTS *pts;
154:   int c;
155:   int i, d;
156:   ARGSET( dummy );
157:   ARYSET(1);
158:   pts=PARTOP(1);
159:   c=IVALUE(2);
160:   Color[0]=BLUE(c);
161:   Color[1]=RED(c);
162:   Color[2]=GREEN(c);
163:   npts[0][0];
164:   if ( pts[0][2]!=OVERSAMPLE ) {
165: #ifdef _GNU_

```

```

166:   asm ( " lea.l _OVERSAMPLE_NOTYET,a1" );
167: #else
168:   #asm
169:   lea.l _OVERSAMPLE_NOTYET,a1
170: #endsasm
171: #endif
172:   return ( 1 );
173: }
174: Direction=0; /* 始めは傾きの大小が不明 */
175: for ( i=1; i<n; i++ ) {
176:   aa_line( pts[i][0], pts[i][1], pts[i+1][0], pts[i+1][1], pts[i][2] );
177: }
178: if ( pts[0][1]==CYCLIC ) { /* 点列が循環している場合、終点と始点をつなぐ */
179:   aa_line( pts[n][0], pts[n][1], pts[1][0], pts[1][1], pts[n][2] );
180: }
181: return ( 0 );
182: }
183: FUNC lines( dummy ) /* アンチエイリアシングなしのバージョン */
184: DUMMY dummy;
185: /* PTS *pts, int c */
186: {
187:   PTS *pts;
188:   int c;
189:   int i, n;
190:   ARGSET( dummy );
191:   ARYSET(1);
192:   pts=PARTOP(1);
193:   c=IVALUE(2);
194:   npts[0][0];
195:   if ( pts[0][2]!=OVERSAMPLE ) {
196: #ifdef _GNU_
197:     asm ( " lea.l _OVERSAMPLE_NOTYET,a1" );
198: #else
199:     #asm
200:     lea.l _OVERSAMPLE_NOTYET,a1
201: #endsasm
202: #endif
203:     return ( 1 );
204:   }
205:   for ( i=1; i<n; i++ ) {
206:     line( PIX(pts[i][0]), PIX(pts[i][1]), PIX(pts[i+1][0]), PIX(pts[i+1][1]), c, 0xFFFF );
207:   }
208:   if ( pts[0][1]==CYCLIC ) {
209:     line( PIX(pts[n][0]), PIX(pts[n][1]), PIX(pts[1][0]), PIX(pts[1][1]), c, 0xFFFF );
210:   }
211:   return ( 0 );
212: }

```

リスト5

```

===== aa_paint.c =====
1: /***** アンチエイリアシング対応イベントルーチン *****/
2: #include <graph.h>
3: #include "anti.h"
4: /* キュー (待ち行列、または FIFO バッファ) */
5: #define MAXQUEUE 1024
6: short Qx[ MAXQUEUE ], Qy[ MAXQUEUE ];
7: int QPi, QPo;
8: #define INIT_QUEUE { QPi=QPo=0; }
9: #define ENQUEUE( X, Y ) { Qx[ QPi ]=(X); Qy[ QPi ]=(Y); QPi=(++QPi)%MAXQUEUE; }
10: #define DEQUEUE( X, Y ) { (X)=Qx[ QPo ]; (Y)=Qy[ QPo ]; QPo=(++QPo)%MAXQUEUE; }
11: #define EMPTY_QUEUE { QPi=QPo }
12: /* タイルおよびトーン (ソリッドスキャンコンバージョンと共有) */
13: extern unsigned char Color[3];
14: extern unsigned char Tile[ N_TILE ][ T_SIZE ][ T_SIZE ]; /*R,G,B の 3色*/
15: extern unsigned int Tile_x[ N_TILE ], Tile_y[ N_TILE ]; /*R,G,B の 3色*/
16: extern unsigned char Tone[ N_TONE ][ T_SIZE ][ T_SIZE ]; /*タイル/パターンの大きさ*/
17: extern unsigned int Tone_x[ N_TILE ], Tone_y[ N_TILE ]; /*タイルの大きさ*/
18: extern int tile_tone_check(); /* タイル・トーンの指定が正しいかどうか調べる */
19: extern unsigned short Slbuf[ N_PIXEL ]; /* 1 スキャンラインぶんフレームバッファ */
20: unsigned short slbuf[512], slbuf[512]; /*ポイント用スキャンラインバッファ*/
21: unsigned char _OUTF_SCREEN[]="指定した座標がスクリーンの範囲外です";
22:
23: FUNC aa_paint( dummy ) /* 関数本体 */
24: DUMMY dummy;
25: /* int x0, y0, c */
26: {
27:   int x0, y0, cmode, c, n_tile, tmode, n_tone;
28:   int tile_x, tile_y, tone_x, tone_y;
29:
30:   int i, x, y, x1, x2;
31:   int sign, sign1;
32:   unsigned int r, g, b, r1, g1, b1, v, vm, s;
33:
34:   ARGSET( dummy );
35:   ARYSET(1);
36:   x0=IVALUE(1);
37:   y0=IVALUE(2);
38:   cmode=IVALUE(3);
39:   o_n_tile=IVALUE(4);
40:   tmode=IVALUE(5);
41:   n_tone=IVALUE(6);
42:   if ( x0<0 || x0>N_PIXEL || y0<0 || y0>N_PIXEL ) { /* 画面外は塗れない */
43: #ifdef _GNU_
44:     asm ( " lea.l _OUTF_SCREEN,a1" );
45: #else
46:     #asm
47:     lea.l _OUTF_SCREEN,a1
48: #endsasm
49: #endif
50:     return ( 1 );
51:   }
52:
53:   if ( cmode==COLOR ) { /* タイルパターンを使わないなら描画色は一定 */
54:     b=BLUE( c );
55:     r=RED( c );
56:     g=GREEN( c );
57:   }
58:   itile_tone_check( cmode, n_tile, &tile_x, &tile_y, tmode, n_tone, &tone_x, &tone_y );
59:   if ( i==1 ) return ( 1 );
60:
61:   INIT_QUEUE; /* FIFO バッファを空にする */
62:   ENQUEUE( x0, y0 ); /* シードを FIFO バッファに入れる */
63:   vm=IVALUE( point( x0, y0 ) ); /* 高与率の最大値は出発点を100%とする */
64:   if ( tmode&ON ) { /* トーンを考慮した高与率の最大値 */
65:     vm=IMAX;
66:   }
67:
68:   while ( !EMPTY_QUEUE ) { /* イベント終了条件 */
69:     DEQUEUE( x0, y0 ); /* シードを FIFO バッファから取り出す */
70:     if ( point( x0, y0 ) &PMASK ) continue; /* イベント済み */
71:     if ( point( x0, y0 ) &PMASK == 0 ) continue; /* 黒いところには塗っても仕方ない */
72:
73:     if ( y0>0 ) /* 上への到達可能性を調べるためのバッファ */
74:       get( 0, y0-1, N_PIXEL-1, y0-1, slbuf, N_PIXEL*sizeof(short) );

```

```

75:   get( 0, y0, N_PIXEL-1, y0, slbuf, N_PIXEL*sizeof(short) );
76:   if ( y0<N_PIXEL-1 ) /* 下への到達可能性を調べるためのバッファ */
77:     get( 0, y0+1, N_PIXEL-1, y0+1, slbuf, N_PIXEL*sizeof(short) );
78:
79:   for ( x=0; x<N_PIXEL; x++ ) {
80:     slbuf[x] &= SLMASK;
81:     slbuf[x] &= SLMASK; /*左右への到達可能性を調べるバッファ*/
82:     if ( slbuf[x]==0 ) slbuf[x]=PMASK; /*黒いところには塗っても仕方ないので*/
83:     if ( slbuf[x]==0 ) slbuf[x]=PMASK; /*始めからイベント済みしておく */
84:     if ( slbuf[x]==0 ) slbuf[x]=PMASK;
85:   }
86:
87:   for ( x1=x0; x1>0; x1-- ) { /* 左へ塗り進める */
88:     if ( cmode==TILE ) { /* タイル/パターン */
89:       bl=Tile[n_tile][y0*tile_y][x1*tile_x][0];
90:       r1=Tile[n_tile][y0*tile_y][x1*tile_x][1];
91:       g1=Tile[n_tile][y0*tile_y][x1*tile_x][2];
92:     }
93:     if ( tmode&ON ) { /* トーンあり */
94:       v=(slbuf[x1]>>VSHIFT)*Tone[n_tone][y0*tone_y][x1*tone_x];
95:     } else { /* トーンなし */
96:       v=slbuf[x1]>>VSHIFT;
97:     }
98:     if ( tmode&TP ) { /* 下地の透けて見える */
99:       s=Slbuf[x1];
100:       b=( BLUE(s)*(vm-v) + bl*v )/vm;
101:       r=( RED(s)*(vm-v) + r1*v )/vm;
102:       g=( GREEN(s)*(vm-v) + g1*v )/vm;
103:     } else { /* 下地は無視 */
104:       b=bl*v/vm;
105:       r=r1*v/vm;
106:       g=g1*v/vm;
107:     }
108:     Slbuf[x1]=RGB( r, g, b, PMASK ); /* イベント済みフラグを立てる */
109:     if ( x1==0 ) break; /* さらに左に進めるか調べる */
110:     if ( slbuf[x1-1]&PMASK ) break; /* イベント済みかどうか調べる */
111:     if ( slbuf[x1]<slbuf[x1-1] ) break; /* イベント済みかどうか調べる */
112:   }
113:   for ( x2=x0; x2<N_PIXEL; x2++ ) { /* 右へ塗り進める、以下同様 */
114:     if ( cmode==TILE ) {
115:       bl=Tile[n_tile][y0*tile_y][x2*tile_x][0];
116:       r1=Tile[n_tile][y0*tile_y][x2*tile_x][1];
117:       g1=Tile[n_tile][y0*tile_y][x2*tile_x][2];
118:     }
119:     if ( tmode&ON ) {
120:       v=(slbuf[x2]>>VSHIFT)*Tone[n_tone][y0*tone_y][x2*tone_x];
121:     } else {
122:       v=slbuf[x2]>>VSHIFT;
123:     }
124:     if ( tmode&TP ) {
125:       s=Slbuf[x2];
126:       b=( BLUE(s)*(vm-v) + bl*v )/vm;
127:       r=( RED(s)*(vm-v) + r1*v )/vm;
128:       g=( GREEN(s)*(vm-v) + g1*v )/vm;
129:     } else {
130:       b=bl*v/vm;
131:       r=r1*v/vm;
132:       g=g1*v/vm;
133:     }
134:     Slbuf[x2]=RGB( r, g, b, PMASK );
135:     if ( x2==N_PIXEL-1 ) break;
136:     if ( slbuf[x2]&PMASK ) break;
137:     if ( slbuf[x2]<slbuf[x2+1] ) break;
138:   }
139:
140:   if ( y0>0 ) { /* 上へ塗り進めるの可能性を調べる */
141:     /*signは傾きの記号 */
142:     /*ひとつ上のスキャンラインの傾きが極大になる(signが + から - に転じる)ところまで塗り進める */
143:     /*塗り進めない傾きの途中で出た場合には、その直前で塗り進める */
144:     sign=1;
145:     for ( x=x1; x<=x2; x++ ) {
146:       if ( slbuf[x]&PMASK || slbuf[x]<slbuf[x] ) {
147:         sign=-1;
148:         continue;
149:       }
150:       if ( x==x2 || slbuf[x+1]&PMASK || slbuf[x+1]<slbuf[x+1] ) {

```



```

150:         if ( sign>0 ) ENGBUE( x, y0-1 );
151:         continue;
152:     }
153:     sign1=sibuf[x+1]-sibuf[x];
154:     if ( sign>0 && sign1<0 ) ENGBUE( x, y0-1 );
155:     if ( sign*sign1<0 ) sign=sign1;
156: }
157: }
158: if ( y0<N_PIXEL-1 ) { /* 下へ塗り進む可能性を調べる、以下同様 */
159:     sign1=1;
160:     for ( x=x1; x<=x2; x++ ) {
161:         if ( sibuf[x]&PMASK || sibuf[x]<sibuf[x] ) {
162:             sign1=1;
163:             continue;

```

```

164:         }
165:         if ( x==x2 || sibuf[x]<=sibuf[x-1]&PMASK || sibuf[x]<=sibuf[x-1] ) {
166:             if ( sign>0 ) ENGBUE( x, y0-1 );
167:             continue;
168:         }
169:         sign1=sibuf[x+1]-sibuf[x];
170:         if ( sign>0 && sign1<0 ) ENGBUE( x, y0-1 );
171:         if ( sign*sign1<0 ) sign=sign1;
172:     }
173: }
174: put( 0, y0, N_PIXEL-1, y0, sibuf, N_PIXEL*sizeof(short));
175: }
176: return ( 0 );
177: }

```

リスト6

***** aa_procs.c *****

```

1: /****** アンチエイリアシング関係の処理 (タイル・トーンなど) *****/
2: #include "anti.h"
3: /* 色 (カラーコードまたはタイルパターン) */
4: unsigned char Color[3]; /* R,G,B の 3色 */
5: unsigned char Tile[ N_TILE ][ T_SIZE ][ T_SIZE ][3]; /* R,G,B の 3色 */
6: unsigned int Tile_x[ N_TILE ], Tile_y[ N_TILE ]; /* タイルパターンの大きさ */
7:
8: /* トーン */
9: unsigned char Tone[ N_TONE ][ T_SIZE ][ T_SIZE ]; /* 単色 */
10: unsigned int Tone_x[ N_TONE ], Tone_y[ N_TONE ]; /* トーンの大きさ */
11: unsigned int Alpha[ N_PIXEL ]; /* 1 スキャンラインぶんの書き率バツファ */
12: unsigned short Sibuf[ N_PIXEL ]; /* 1 スキャンラインぶんのフレームバッファ */
13: unsigned short Temp[ T_SIZE+T_SIZE ]; /* tile_get(), tone_get() 用のテンポラリ配列 */
14: unsigned char TOOMANY_PATTERN[ ]="パターン番号が大きすぎます";
15: unsigned char OVERSIZE[ ]="パターンのサイズが大きすぎます";
16:
17: FUNC tile_get( dummy )
18: DUMMY dummy;
19: /* int n, int x1, int y1, int x2, int y2 */
20: {
21:     int n, x1, y1, x2, y2;
22:     int i, j, dx, dy, ax, ay, x, y;
23:     unsigned short c;
24:     ARGSET( dummy );
25:     n=IVALUE(1);
26:     x1=IVALUE(2);
27:     y1=IVALUE(3);
28:     x2=IVALUE(4);
29:     y2=IVALUE(5);
30:     if ( n>N_TILE ) {
31:         #ifdef _GNUC_
32:             asm( " lea.l _TOOMANY_PATTERN,a1" );
33:         #else
34:             #asm
35:             lea.l _TOOMANY_PATTERN,a1
36:             #endasm
37:         #endif
38:         return ( 1 );
39:     }
40:     dx = ABS( x2-x1 )+1;
41:     dy = ABS( y2-y1 )+1;
42:     ax = SIGN( x2-x1 );
43:     ay = SIGN( y2-y1 );
44:     if ( dx>T_SIZE || dy>T_SIZE ) {
45:         #ifdef _GNUC_
46:             asm( " lea.l _OVERSIZE,a1" );
47:         #else
48:             #asm
49:             lea.l _OVERSIZE,a1
50:             #endasm
51:         #endif
52:         return ( 1 );
53:     }
54:     Tile_x[n] = dx;
55:     Tile_y[n] = dy;
56:     get( MIN(x1,x2), MIN(y1,y2), MAX(x1,x2), MAX(y1,y2), Temp, dx*dy*sizeof(short) );
57:     for ( i=0; y=(sy>0)?(0):(dy-1); i+=dx; i++, y+=ay ) {
58:         for ( j=0; x=(sx>0)?(0):(dx-1); j+=dx; j++, x+=ax ) {
59:             c=Temp[j+dx*x];
60:             Tile[n][i][j][0]=BLUE(c);
61:             Tile[n][i][j][1]=RED(c);
62:             Tile[n][i][j][2]=GREEN(c);
63:         }
64:     }
65:     return ( 0 );
66: }
67:
68: FUNC tone_get( dummy )
69: DUMMY dummy;
70: /* int n, int x1, int y1, int x2, int y2 */
71: {
72:     int n, x1, y1, x2, y2;
73:     int i, j, dx, dy, ax, ay, x, y;
74:     unsigned short c;
75:
76:     ARGSET( dummy );
77:     n=IVALUE(1);
78:     x1=IVALUE(2);
79:     y1=IVALUE(3);
80:     x2=IVALUE(4);
81:     y2=IVALUE(5);
82:     if ( n>N_TONE ) {
83:         #ifdef _GNUC_
84:             asm( " lea.l _TOOMANY_PATTERN,a1" );
85:         #else
86:             #asm
87:             lea.l _TOOMANY_PATTERN,a1
88:             #endasm
89:         #endif
90:         return ( 1 );
91:     }
92: }
93:
94: dx = ABS( x2-x1 )+1;
95: dy = ABS( y2-y1 )+1;
96: sx = SIGN( x2-x1 );
97: sy = SIGN( y2-y1 );
98:
99: if ( dx>T_SIZE || dy>T_SIZE ) {
100:     #ifdef _GNUC_
101:         asm( " lea.l _OVERSIZE,a1" );
102:     #else
103:         #asm
104:         lea.l _OVERSIZE,a1
105:         #endasm
106:     #endif
107:     return ( 1 );
108: }

```

```

109: Tone_x[n] = dx;
110: Tone_y[n] = dy;
111: get( MIN(x1,x2), MIN(y1,y2), MAX(x1,x2), MAX(y1,y2), Sibuf, dx*dy*sizeof(short) );
112: for ( i=0; y=(sy>0)?(0):(dy-1); i+=dx; i++, y+=ay ) {
113:     for ( j=0; x=(sx>0)?(0):(dx-1); j+=dx; j++, x+=ax ) {
114:         c=Temp[j+dx*x];
115:         Tone[n][i][j]=IVALUE(c);
116:     }
117: }
118: return ( 0 );
119: }
120: unsigned char ILLEGAL_NTILE[ ]="タイルパターン番号が不正です";
121: unsigned char TILE_TOO_LARGE[ ]="タイルパターンのサイズが大きすぎます";
122: unsigned char ILLEGAL_CNMODE[ ]="色タイルのモードを正しく指定して下さい";
123: unsigned char ILLEGAL_NTONE[ ]="色トーンの番号が不正です";
124: unsigned char TONE_TOO_LARGE[ ]="色トーンのサイズが大きすぎます";
125: unsigned char ILLEGAL_TMODE[ ]="色トーンのモードを正しく指定して下さい";
126: int tile_tone_check( cmode, n_tile, tile_x, tile_y, tone_x, tone_y )
127: int cmode, n_tile, *tile_x, *tile_y, *tone_x, *tone_y;
128: {
129:     switch ( cmode ) {
130:         case COLOR:
131:             break;
132:         case TILE:
133:             if ( n_tile > N_TILE ) {
134:                 #ifdef _GNUC_
135:                     asm( " lea.l _ILLEGAL_NTILE,a1" );
136:                 #else
137:                     #asm
138:                     lea.l _ILLEGAL_NTILE,a1
139:                     #endasm
140:                 #endif
141:                 return ( 1 );
142:             }
143:             *tile_x=Tile_x[n_tile];
144:             *tile_y=Tile_y[n_tile];
145:             if ( *tile_x>T_SIZE || *tile_y>T_SIZE ) {
146:                 #ifdef _GNUC_
147:                     asm( " lea.l _TILE_TOO_LARGE,a1" );
148:                 #else
149:                     #asm
150:                     lea.l _TILE_TOO_LARGE,a1
151:                     #endasm
152:                 #endif
153:                 return ( 1 );
154:             }
155:             break;
156:         default:
157:             #ifdef _GNUC_
158:                 asm( " lea.l _ILLEGAL_CNMODE,a1" );
159:             #else
160:                 #asm
161:                 lea.l _ILLEGAL_CNMODE,a1
162:                 #endasm
163:             #endif
164:             return ( 1 );
165:             break;
166:     }
167:     switch ( tmode ) {
168:         case ON_TP:
169:             case ON_NTP:
170:                 if ( n_tone > N_TONE ) {
171:                     #ifdef _GNUC_
172:                         asm( " lea.l _ILLEGAL_NTONE,a1" );
173:                     #else
174:                         #asm
175:                         lea.l _ILLEGAL_NTONE,a1
176:                         #endasm
177:                     #endif
178:                     return ( 1 );
179:                 }
180:                 *tone_x=Tone_x[n_tone];
181:                 *tone_y=Tone_y[n_tone];
182:                 if ( *tone_x>T_SIZE || *tone_y>T_SIZE ) {
183:                     #ifdef _GNUC_
184:                         asm( " lea.l _TONE_TOO_LARGE,a1" );
185:                     #else
186:                         #asm
187:                         lea.l _TONE_TOO_LARGE,a1
188:                         #endasm
189:                     #endif
190:                     return ( 1 );
191:                 }
192:                 break;
193:             case OFF_TP:
194:             case OFF_NTP:
195:                 break;
196:             default:
197:                 #ifdef _GNUC_
198:                     asm( " lea.l _ILLEGAL_TMODE,a1" );
199:                 #else
200:                     #asm
201:                     lea.l _ILLEGAL_TMODE,a1
202:                     #endasm
203:                 #endif
204:                 return ( 1 );
205:                 break;
206:     }
207:     return ( 0 );
208: }
209:
210: FUNC whitespace()
211: {
212:     int i;
213:     for ( i=0; i<N_PIXEL; i++ ) {
214:         Sibuf[i]=RGBI( IMAX, IMAX, IMAX, 0 );
215:     }
216:     for ( i=0; i<N_PIXEL; i++ ) {
217:         put( 0, i, N_PIXEL-1, i, Sibuf, N_PIXEL*sizeof(short) );

```



```

218: }
219: return ( 0 );
220: }
221:
222: FUNC reverse()
223: {
224:   int i, j;
225:   for ( i=0; i<N_PIXEL; i++ ) {
226:     get( 0, i, N_PIXEL-1, i, Sibuf, N_PIXEL*sizeof(short) );
227:     for ( j=0; j<N_PIXEL; j++ ) {
228:       Sibuf[j] ^= RGB( IMAX, IMAX, IMAX, 0 );
229:     }
230:     put( 0, i, N_PIXEL-1, i, Sibuf, N_PIXEL*sizeof(short) );
231:   }
232:   return ( 0 );
233: }
234:
235: FUNC maskclear()
236: {
237:   int i, j;
238:   for ( i=0; i<N_PIXEL; i++ ) {
239:     get( 0, i, N_PIXEL-1, i, Sibuf, N_PIXEL*sizeof(short) );
240:     for ( j=0; j<N_PIXEL; j++ ) {
241:       Sibuf[j] ^= RGB( IMAX, IMAX, IMAX, 0 );
242:     }
243:     put( 0, i, N_PIXEL-1, i, Sibuf, N_PIXEL*sizeof(short) );
244:   }
245:   return ( 0 );
246: }
247:
248: FUNC monotone()

```

```

249:   int i, j;
250:   unsigned int s, c;
251:   for ( i=0; i<N_PIXEL; i++ ) {
252:     get( 0, i, N_PIXEL-1, i, Sibuf, N_PIXEL*sizeof(short) );
253:     for ( j=0; j<N_PIXEL; j++ ) {
254:       s=Sibuf[j];
255:       c=(RED(s)*77+GREEN(s)*151+BLUE(s)*28)/256;
256:       Sibuf[j] = RGB( c, c, c );
257:     }
258:     put( 0, i, N_PIXEL-1, i, Sibuf, N_PIXEL*sizeof(short) );
259:   }
260:   return ( 0 );
261: }

```

リスト7

```

===== main.c =====
1: /****** パラメータ受け渡し用変数の実体 *****/
2:
3: unsigned short *par; /* 一時的な引数リスト */
4: unsigned short *ary[10+1]; /* 一時的な配列リスト; X-BASIC の引数は最大 10 個 */
5:
6: /****** コンパイラを通すためのダミー (実行されない) *****/
7:
8: void main()
9: {
10: }

```

リスト8

```

===== pts_curve.c =====
1: /****** Bezier曲線を使った内挿により自由曲線を発生する *****/
2: #include <math.h>
3: #include "anti.h"
4: #define MAXPTS1 256 /* 入力点列の長さの最大値 */
5: #define SCALE 32 /* 整数演算の精度を確保するための倍率 */
6: #define MIN_LENGTH (OVERSAMPLE*SCALE) /* 再分割を打ち切る制御点の距離 */
7: typedef int vector[2]; /* 整数値ベクトル */
8: typedef double vector2[2]; /* 実数値ベクトル */
9: ivector Pstemp[MAXPTS1][3]; /* テンポラリの制御点 */
10: int Type_pts, N_pts1, N_pts2, Maxpts2;
11: PTS *pts2;
12: #define COPY( V1, V2 ) { V1[0]=V2[0]; V1[1]=V2[1]; } /* ベクトルのコピー */
13: #define SCOPY( V1, V2 ) { V1[0]=V2[0]/SCALE; V1[1]=V2[1]/SCALE; } /* 倍率を加味したベクトルのコピー */
14: #define LENGTH( V ) { sqrt( V[0]*V[0]+V[1]*V[1] ) } /* ベクトルの長さ */
15: void normalize( v1, v2 ) /* 単位ベクトル化 */
16: vector v1, v2;
17: {
18:   double l;
19:   l=LENGTH( v1 );
20:   v2[0] = v1[0]/l;
21:   v2[1] = v1[1]/l;
22:   return;
23: }
24: void mult_factor_vec( v1, v2 ) /* 方向は変えないで長さを同じにする */
25: vector v1, v2;
26: {
27:   double factor;
28:   factor=LENGTH( v2 )/LENGTH( v1 );
29:   v2[0] = factor*v1[0];
30:   v2[1] = factor*v1[1];
31:   return;
32: }
33: void control( p1, p2, p3 ) /* サンプル点から制御点を発生する */
34: ivector p1, p2, p3;
35: {
36:   vector va, vb, vc, norma, normb;
37:   double la, lb;
38:   int i, j, k;
39:   va[0] = (double)(p2[0]-p1[0]); /* 隣のスランポイントへの方向ベクトル */
40:   va[1] = (double)(p2[1]-p1[1]);
41:   vb[0] = (double)(p3[0]-p2[0]);
42:   vb[1] = (double)(p3[1]-p2[1]);
43:   normalize( va, norma ); /* 単位ベクトル化 */
44:   normalize( vb, normb );
45:   vc[0] = { norma[0]*normb[0] }/2.0; /* 2等分線を取ると */
46:   vc[1] = { norma[1]*normb[1] }/2.0; /* p2 における接線ベクトル */
47:   mult_factor_vec( va, va );
48:   mult_factor_vec( vb, vb );
49:   p1[0] = p2[0]-(int)va[0]; /* p2 の隣2つの制御点 */
50:   p1[1] = p2[1]-(int)va[1];
51:   p3[0] = p2[0]+(int)vb[0];
52:   p3[1] = p2[1]+(int)vb[1];
53:   return;
54: }
55:
56: int bezier( s1, s2, s3, s4 ) /* 4つの制御点から Bezier 曲線を発生する */
57: ivector s1, s2, s3, s4;
58: {
59:   ivector s12, s23, s34;
60:   double lx, ly;
61:   #define s1234 s23
62:   #define s123 s2
63:   #define s234 s3
64:   lx=(double){s4[0]-s1[0]}; /* 制御点間の距離が十分短くなったら */
65:   ly=(double){s4[1]-s1[1]}; /* 距離を打ち切る */
66:   if ( (lx*lx+ly*ly)<((double)MIN_LENGTH*(double)MIN_LENGTH) ) {
67:     if ( ++N_pts2>Maxpts2 ) return ( 1 );
68:     SCOPY( pts2[N_pts2], s2 ); /* 曲線を微小線分で近似する */
69:     if ( ++N_pts2>Maxpts2 ) return ( 1 );
70:     SCOPY( pts2[N_pts2], s3 );
71:     if ( ++N_pts2>Maxpts2 ) return ( 1 );
72:     SCOPY( pts2[N_pts2], s4 );
73:     return ( 0 );
74:   }
75:
76:   #define MID( V1, V2, V12 ) { V12[0]=(V1[0]+V2[0])/2; V12[1]=(V1[1]+V2[1])/2; }
77:   MID( s1, s2, s12 ); /* 中点を取っていく */
78:   MID( s2, s3, s23 );
79:   MID( s3, s4, s34 );
80:   MID( s12, s23, s123 );
81:   MID( s23, s34, s234 );
82:   MID( s123, s234, s1234 );
83:   if ( bezier( s1, s12, s123, s1234 )!=0 ) return ( 1 ); /* 再分割 */
84:   if ( bezier( s1234, s234, s34, s4 )!=0 ) return ( 1 ); /* 出力点列が不足すればエラー */
85:
86:   #undef s1234
87:   #undef s123
88:   #undef s234
89:
90:   return ( 0 );

```

```

91:
92: }
93:
94: extern unsigned char OVERSAMPLE_NOTYET[];
95: unsigned char CURVE_TOO_MANY[] = "入力点の数が多すぎます";
96: unsigned char CURVE_EXHAUSTED[] = "出力の配列の大きさが足りません";
97:
98: FUNC pts_curve( dummy ) /* 関数本体 */
99: DUMMY dummy;
100: /* PTS *pts1, int w1, int w2, PTS *pts2 */
101: {
102:   PTS *pts1;
103:   int w1, w2;
104:   int i, j, e, n1, n2, m;
105:   ARGSET( dummy );
106:   ARGSET(1);
107:   pts1=PARTOP(1);
108:   w1=VALUE(2);
109:   w2=VALUE(3);
110:   ARGSET(4);
111:   pts2=PARTOP(4);
112:   Maxpts2=1;
113:   for ( i=0; i<DIM(4); i++ ) {
114:     Maxpts2 += ( SUFFIX(4,i+1)+1 );
115:   }
116:   Maxpts2 /= FISSIZE; /* 出力点列の長さの最大値 */
117:
118:   N_pts1=pts1[0][0]; /* 入力点列の長さ */
119:   Type_pts=pts1[0][1]; /* 入力点列のタイプ */
120:   if ( N_pts1>MAXPTS1 ) {
121:     #ifdef _GNUC
122:     asm ( "lea.l _CURVE_TOO_MANY,a1" );
123:     #else
124:     #asm
125:     lea.l _CURVE_TOO_MANY,a1
126:     #endasm
127:     #endif
128:     return ( 1 );
129:   }
130:   if ( pts1[0][2]!=OVERSAMPLE ) {
131:     #ifdef _GNUC
132:     asm ( "lea.l _OVERSAMPLE_NOTYET,a1" );
133:     #else
134:     #asm
135:     lea.l _OVERSAMPLE_NOTYET,a1
136:     #endasm
137:     #endif
138:     return ( 1 );
139:   }
140:   for ( i=1; i<N_pts1; i++ ) {
141:     Pstemp[i][0][0]=pts1[i][0]*SCALE; /* サンプル点を制御点 */
142:     Pstemp[i][0][1]=pts1[i][1]*SCALE;
143:   }
144:   /* 制御点の処理 ... サンプル点間を3等分する */
145:   for ( i=1; i<N_pts1; i++ ) {
146:     Pstemp[i][1][0]=(Pstemp[i][0][0]*2+Pstemp[i+1][0][0])/3;
147:     Pstemp[i][1][1]=(Pstemp[i][0][1]*2+Pstemp[i+1][0][1])/3;
148:     Pstemp[i][2][0]=(Pstemp[i][0][0]+Pstemp[i+1][0][0]*2)/3;
149:     Pstemp[i][2][1]=(Pstemp[i][0][1]+Pstemp[i+1][0][1]*2)/3;
150:   }
151:   if ( Type_pts==CYCLIC ) {
152:     Pstemp[N_pts1][1][0]=(Pstemp[N_pts1][0][0]*2+Pstemp[1][0][0])/3;
153:     Pstemp[N_pts1][1][1]=(Pstemp[N_pts1][0][1]*2+Pstemp[1][0][1])/3;
154:     Pstemp[N_pts1][2][0]=(Pstemp[N_pts1][0][0]+Pstemp[1][0][0]*2)/3;
155:     Pstemp[N_pts1][2][1]=(Pstemp[N_pts1][0][1]+Pstemp[1][0][1]*2)/3;
156:   }
157:   /* 制御点の発生 */
158:   for ( i=2; i<N_pts1; i++ ) {
159:     control( Pstemp[i-1][2], Pstemp[i][0], Pstemp[i][1] );
160:   }
161:   if ( Type_pts==CYCLIC ) {
162:     control( Pstemp[N_pts1-1][2], Pstemp[N_pts1][0], Pstemp[N_pts1][1] );
163:     control( Pstemp[N_pts1][2], Pstemp[1][0], Pstemp[1][1] );
164:   }
165:   /* 自由曲線の生成 */
166:   SCOPY( pts2[1], Pstemp[1][0] ); /* 始点はマニュアルでコピー */
167:   if ( w1<0 || w2<0 ) {
168:     m=1; /* 線の太さを点列レベルで (頂点ごとに) 指定されている */
169:   } else {
170:     m=0; /* 線の太さはコマンドレベル (点列全体) で指定されている */
171:   }
172:   N_pts2=1;
173:   for ( i=1; i<N_pts1; i++ ) {
174:     n1=N_pts2;
175:     e=bezier( Pstemp[i][0], Pstemp[i][1], Pstemp[i][2], Pstemp[i+1][0] );
176:     n2=N_pts2;
177:     if ( m ) {
178:       for ( j=n1; j<n2; j++ ) {
179:         pts2[j][2]=(pts1[i][2]*(n2-j)+pts1[i+1][2]*(j-n1))/(n2-n1);
180:       }
181:     }
182:   }

```



```

183: if ( e==0 && Type_pts==CYCLIC ) { /* 点列が閉環する場合 */
184:     n1=N_pts2;
185:     e=bezier( Pstatemp[N_pts][0], Pstatemp[N_pts][1], Pstatemp[N_pts][2], Pstatemp[1][0] );
186:     n2=N_pts2;
187:     N_pts2--; /* 終点は始点と一致するので捨てる */
188:     if ( n1 ) {
189:         for ( j=n1; j<n2; j++ ) {
190:             pts2[j][2]=(pts1[j][2]*(n2-j)+pts1[1][2]*(j-n1))/(n2-n1);
191:         }
192:     }
193: }
194: if ( e!=0 ) {
195: #ifdef _GNUC_
196:     asm ( " lea.l _CURVE_EXHAUSTED,a1" );
197: #else
198:     #asm

```

```

199:     lea.l _CURVE_EXHAUSTED,a1
200: #endif
201: #endif
202:     return ( 1 );
203: }
204: pts2[0][0]=N_pts2;
205: pts2[0][1]=Type_pts;
206: pts2[0][2]=OVERSAMPLE;
207: if ( e==0 ) {
208:     for ( i=1; i<N_pts2; i++ ) {
209:         pts2[i][2]=(pts1[i][2]*(N_pts2-i)+pts1[0][2]*(i-1))/(N_pts2-1);
210:     }
211: }
212: return ( 0 );
213: }

```

リスト9

```

===== pts_procs.c =====
1: /****** 点列の移動および接続 *****/
2:
3: #include "anti.h"
4:
5: unsigned char OVERSAMPLE_NOTYET[]="オーバーサンプリング座標に変換してください";
6:
7: unsigned char MOVE_INCOMPATIBLE[]="移動先の点列とサイズが合いません";
8:
9: FUNC pts_move( dummy )
10: DUMMY dummy;
11: /* PTS *pts1, int x, int y, PTS *pts2 */
12: {
13:     PTS *pts1, *pts2;
14:     int x, y, i;
15:     int n1, n2;
16:
17:     ARGSET( dummy );
18:     ARGSET(1);
19:     pts1=PARTOP(1);
20:     x=IVALUE(2);
21:     y=IVALUE(3);
22:     ARGSET(4);
23:     pts2=PARTOP(4);
24:     n2=1;
25:     for ( i=0; i<DIM(4); i++ ) {
26:         n2 += ( SUFFIX(4,i+1)+1 );
27:     }
28:     n2 /= PTSSIZE;
29:     if ( pts1[0][2]!=OVERSAMPLE ) {
30: #ifdef _GNUC_
31:         asm ( " lea.l _OVERSAMPLE_NOTYET,a1" );
32: #else
33:         #asm
34:         lea.l _OVERSAMPLE_NOTYET,a1
35: #endif
36: #endif
37:     return ( 1 );
38: }
39: n1=pts1[0][0];
40: if ( (n1+i)*n2 ) {
41: #ifdef _GNUC_
42:     asm ( " lea.l _MOVE_INCOMPATIBLE,a1" );
43: #else
44:     #asm
45:     lea.l _MOVE_INCOMPATIBLE,a1
46: #endif
47: #endif
48:     return ( 1 );
49: }
50: pts2[0][0]=n1;
51: pts2[0][1]=pts1[0][1];
52: pts2[0][2]=OVERSAMPLE;
53: for ( i=1; i<n1; i++ ) {
54:     pts2[i][0]=pts1[i][0]*x;
55:     pts2[i][1]=pts1[i][1]*y;
56:     pts2[i][2]=pts1[i][2];
57: }
58: return ( 0 );
59: }
60:
61: unsigned char APPEND_INSUFFICIENT[]="移動先の点列のサイズが足りません";
62:
63: FUNC pts_append( dummy )
64: DUMMY dummy;
65: /* PTS *pts1, int x, int y, PTS *pts2 */
66: {
67:     PTS *pts1, *pts2;
68:     int x, y, i;
69:     int n1, n2;
70:
71:     ARGSET( dummy );
72:     ARGSET(1);
73:     pts1=PARTOP(1);

```

```

74:     ARGSET(2);
75:     pts2=PARTOP(2);
76:     if ( pts1[0][2]!=OVERSAMPLE ) {
77: #ifdef _GNUC_
78:         asm ( " lea.l _OVERSAMPLE_NOTYET,a1" );
79: #else
80:         #asm
81:         lea.l _OVERSAMPLE_NOTYET,a1
82: #endif
83: #endif
84:     return ( 1 );
85: }
86: n1=1;
87: for ( i=0; i<DIM(1); i++ ) {
88:     n1 += ( SUFFIX(1,i+1)+1 );
89: }
90: n1 /= PTSSIZE;
91: n2=pts2[0][0];
92: if ( n1<(pts1[0][0]+n2) ) {
93: #ifdef _GNUC_
94:     asm ( " lea.l _APPEND_INSUFFICIENT,a1" );
95: #else
96:     #asm
97:     lea.l _APPEND_INSUFFICIENT,a1
98: #endif
99: #endif
100:     return ( 1 );
101: }
102: n1=pts1[0][0];
103: pts1[0][0]=n1+n2-1;
104: x=pts1[n1][0]-pts2[1][0];
105: y=pts1[n1][1]-pts2[1][1];
106: for ( i=1; i<n2; i++ ) {
107:     pts1[i+n1][0]=pts2[i+1][0]+x;
108:     pts1[i+n1][1]=pts2[i+1][1]+y;
109:     pts1[i+n1][2]=pts2[i+1][2];
110: }
111: return ( 0 );
112: }
113:
114: unsigned char OVERSAMPLE_ALREADY[]="オーバーサンプリング済み";
115:
116: FUNC pts_oversample( dummy )
117: DUMMY dummy;
118: /* PTS *pts */
119: {
120:     PTS *pts;
121:     int n, i;
122:
123:     ARGSET( dummy );
124:     ARGSET(1);
125:     pts=PARTOP(1);
126:     if ( pts[0][2]!=OVERSAMPLE ) {
127: #ifdef _GNUC_
128:         asm ( " lea.l _OVERSAMPLE_ALREADY,a1" );
129: #else
130:         #asm
131:         lea.l _OVERSAMPLE_ALREADY,a1
132: #endif
133: #endif
134:     return ( 1 );
135: }
136: pts[0][2]=OVERSAMPLE;
137: n=pts[0][0];
138: for ( i=1; i<n; i++ ) {
139:     pts[i][0] = OVER( pts[i][0] );
140:     pts[i][1] = OVER( pts[i][1] );
141: }
142: return ( 0 );
143: }

```

リスト10

```

===== anti.h =====
1: /****** 汎用マクロなどの定義 *****/
2:
3: #define PTSSIZE 3 /* 輪郭を点列で表現する */
4: typedef int PTS[ PTSSIZE ];
5:
6: #define N_PIXEL 512 /* スクリーンのサイズは 512x512 ピクセル */
7:
8: #define OVERSAMPLE 8 /* オーバーサンプリング倍率 */
9: #define OVER2 (OVERSAMPLE*OVERSAMPLE) /* 1ピクセルあたりのサブピクセル数 */
10:
11: /* 通常の座標からオーバーサンプリング座標に変換する、など */
12: #define OVER( X ) ((X)*OVERSAMPLE*(OVERSAMPLE/2))
13: #define PIX( X ) ((X)/OVERSAMPLE)
14: #define SUBPIX( X ) ((X)/OVERSAMPLE)
15:
16: /* 点列のフォーマット *****/
17:
18: PTS *pts; /* 宣言してあるとする */
19:
20: pts[0][0] 点列を構成する点の数
21: pts[0][1] 点列のタイプ (片道通行か循環しているか)
22: pts[0][2] オーバーサンプリング倍率 (ここが OVERSAMPLE でないなら描画関数はエラーになる)
23:
24: ** 第 i 点の情報 ( 1 ≤ i ≤ pts[0][0] ) **
25:
26: pts[i][0] x 座標

```

```

27: pts[i][1] y 座標
28: pts[i][2] 線の幅 (この値は OVERSAMPLE なら 2のべき乗になる)
29:
30: *****/
31:
32: /* 点列のタイプ: 片道通行か循環しているか */
33:
34: #define NORMAL 0
35: #define CYCLIC 1
36:
37:
38: /* X-BASIC からの引数をアクセスする */
39:
40: typedef int FUNC; /* X-BASIC のマクロ関数: 戻り値はエナコード */
41: typedef int DUMMY; /* C からの引数の受け直しの違いを格納するダミー引数 */
42:
43: extern unsigned short *par; /* 一時的な変数リスト */
44: extern unsigned short *ary[10+1]; /* 一時的な変数リスト: X-BASIC の引数は最大 10 個 */
45:
46: #define ARGSET( A ) ( par[(unsigned short *)AA] ) /* 引数 */
47: #define ATOP( I ) ( (I)+4 ) /* 第 I 引数の先頭 */
48: #define TYPE( I ) ( par[ATOP(I)] ) /* 点列の型 */
49: #define IVALUE( I ) ( par[ATOP(I)+3]<16 ? par[ATOP(I)+4] : 1 ) /* int の値 */
50: #define CVALUE( I ) ( par[ATOP(I)+4] ) /* char の値 */
51: #define ARGSET( I ) ( ary[(I)-(unsigned short *)IVALUE(I)] ) /* 配列 */
52: #define DIM( I ) ( ary[(I)-(unsigned short *)IVALUE(I)] ) /* 配列の次元 */
53: #define ELEMENT( I ) ( ary[(I)-(unsigned short *)IVALUE(I)] ) /* 配列要素のサイズ */

```



```

54: #define SUFFIX( I, J ) ( ary[I][J+3] ) /*第 J 添字の最大値*/
55: #define GETTOP( I ) ( &ary[I][IDIM(I)+3+2] ) /*配列の先頭*/
56: #define LARTOP( I ) ( (int *)ARYTOP(I) ) /*int 配列の先頭*/
57: #define CARYTOP( I ) ( (unsigned char *)ARYTOP(I) ) /*char 配列の先頭*/
58: #define PARYTOP( I ) ( (PTS *)ARYTOP(I) ) /*PTS 配列の先頭*/
59:
60: /* その他、便利なマクロ */
61:
62: #define ABS( X ) (((X)>0)?(X):(-(X))) /* X の絶対値 */
63: #define SQN( X ) (((X)>0)?1:((X)<0)?(-1):0) /* X の符号 (正負または零) */
64:
65: #define MIN( X, Y ) (((X)>(Y))?(Y):(X)) /* X,Y のうち小さいほう */
66: #define MAX( X, Y ) (((X)>(Y))?(X):(Y)) /* X,Y のうち大きいほう */
67:
68: /* アンチエイリアシング関係 ... ペイントおよびソリッドスキャンコンバージョン */
69:
70: #define N_TILE 8 /* 格納できるタイルパターンの数 */
71: #define N_TONE 8 /* 格納できるトーンパターンの数 */
72: #define T_SIZE 64 /* タイル及びトーンパターンの大きさ */
73:
74: #define COLOR 0 /* 単色で塗り潰す */
75: #define TILE 1 /* タイルパターンにしたがって色をつける */
76:
77: #define OFF_NTP 0 /* トーンは使わない・ベタ塗り */
78: #define ON_NTP 1 /* トーンを使う・ベタ塗り */
79: #define OFF_TP 2 /* トーンは使わない・下地は透ける */
80: #define ON_TP 3 /* トーンを使う・下地は透ける */
81: #define ON 1 /* トーンを使う */
82: #define TP 2 /* 下地は透ける */

```

```

83:
84: /* R,G,B ごとの輝度を得るためのマスクとビットシフト */
85: #define VMASK_B 62 /* 0b00000000000111110 */
86: #define VMASK_R 1984 /* 0b0000011111000000 */
87: #define VMASK_G 53488 /* 0b1111000000000000 */
88:
89: #define SHIFT_B 1
90: #define SHIFT_R 6
91: #define SHIFT_G 11
92:
93: /* 輝度ビットの値を取り出すためのマスク */
94: #define VMASK_I 1 /* 0b0000000000000001 */
95: #define VSHIFT VMASK_R /* 輝度の代表値は赤チャンネルから取ってくる */
96: #define VSHIFT VMASK_I /* ペイント済みフラグには輝度ビットを用いる */
97: #define FMASK VMASK_I
98: #define SLMASK (FMASK|VMASK)
99:
100: /* R,G,B および R,G,B,I からカラーコードを計算する */
101: #define RGB(R,G,B) ((B)<<SHIFT_B|(R)<<SHIFT_R|(G)<<SHIFT_G)
102: #define RGBI(R,G,B,I) ((B)<<SHIFT_B|(R)<<SHIFT_R|(G)<<SHIFT_G|(I))
103:
104: /* カラーコードから R,G,B,I 成分を取り出す */
105: #define BLUE(C) (((C)&VMASK_B)>>SHIFT_B)
106: #define RED(C) (((C)&VMASK_R)>>SHIFT_R)
107: #define GREEN(C) (((C)&VMASK_G)>>SHIFT_G)
108: #define INTENSITY(C) ((C)&VMASK_I)
109: #define VALUE(C) (((C)&VMASK)>>VSHIFT)
110: #define IMAX 31 /* R,G,B の輝度の最大値 */

```

リスト11

***** anti.s *****

```

1: ***** 外部関数ヘッダ *****
2: # pts_curve.c
3: # pts_curve( PTS *pts1, int w1, int w2, PTS *pts2 )
4: # pts_proc.c
5: # pts_append( PTS *pts1, PTS *pts2 )
6: # pts_move( PTS *pts1, int x, int y, PTS *pts2 )
7: # pts_oversample( PTS *pts )
8: # aa_lines.c
9: # aa_lines( PTS *pts, int c )
10: # lines( PTS *pts, int c )
11: # aa_scanconv.c
12: # aa_scanconv( PTS *pts, int cmode, int c/n_tile, int tmode, int n_tone )
13: # scanconv( PTS *pts, int c )
14: # aa_paint.c
15: # aa_paint( int x, int y, int cmode, int c/n_tile, int tmode, int n_tone )
16: # aa_proc.c
17: # tile_get( int n, int x1, int y1, int x2, int y2 )
18: # tone_get( int n, int x1, int y1, int x2, int y2 )
19: # whitepaper( void )
20: # reverse( void )
21: # maskclear( void )
22: # monotone( void )
23: * インフォメーション・テーブル
24: do.l X_INIT
25: do.l X_RUN
26: do.l X_END
27: do.l X_SYS
28: do.l X_BRK
29: do.l X_CTRL_D
30: do.l X_RES1
31: do.l X_RES2
32: do.l PTR_TOKEN
33: do.l PTR_PARAM
34: do.l PTR_EXEC
35: do.l 0,0,0,0,0
36: X_INIT:
37: X_RUN:
38: X_END:
39: X_SYS:
40: X_BRK:
41: X_CTRL_D:
42: X_RES1:
43: X_RES2:
44: rts
45:
46: * 関数名テーブル
47: PTR_TOKEN:
48: do.b 'pts_curve',0
49: do.b 'pts_append',0
50: do.b 'pts_move',0
51: do.b 'pts_oversample',0
52: do.b 'lines',0
53: do.b 'aa_lines',0
54: do.b 'scanconv',0
55: do.b 'aa_scanconv',0
56: do.b 'aa_paint',0
57: do.b 'tile_get',0
58: do.b 'tone_get',0
59: do.b 'whitepaper',0
60: do.b 'reverse',0
61: do.b 'maskclear',0
62: do.b 'monotone',0
63: do.b 0
64: .even
65:
66: * パラメータ・テーブルへのポインタ
67: PTR_PARAM:
68: do.l PTS_CURVE_PAR
69: do.l PTS_APPEND_PAR
70: do.l PTS_MOVE_PAR
71: do.l PTS_OVERSAMPLE_PAR
72: do.l LINES_PAR
73: do.l AA_LINES_PAR
74: do.l SCANCONV_PAR
75: do.l AA_SCANCONV_PAR
76: do.l AA_PAINT_PAR
77: do.l TILE_GET_PAR
78: do.l TONE_GET_PAR
79: do.l WHITEPAPER_PAR
80: do.l REVERSE_PAR
81: do.l MASKCLEAR_PAR
82: do.l MONOTONE_PAR
83:
84:
85: * パラメータ・テーブル
86:
87: int_val: equ $0002 /* int
88: PTS_ary: equ $0052 /* 1D-array of PTS ( 2D-array of int )
89: float_ary: equ $0037 /* 1D-array of float,int,char
90: void_ret: equ $ffff /* void
91:
92: PTS_CURVE_PAR:

```

```

93: do.w PTS_ary
94: do.w int_val
95: do.w int_val
96: do.w PTS_ary
97: do.w void_ret
98: PTS_APPEND_PAR:
99: do.w PTS_ary
100: do.w PTS_ary
101: do.w void_ret
102: PTS_MOVE_PAR:
103: do.w PTS_ary
104: do.w int_val
105: do.w int_val
106: do.w PTS_ary
107: do.w void_ret
108: PTS_OVERSAMPLE_PAR:
109: do.w PTS_ary
110: do.w void_ret
111: LINES_PAR:
112: do.w PTS_ary
113: do.w int_val
114: do.w void_ret
115: AA_LINES_PAR:
116: do.w PTS_ary
117: do.w int_val
118: do.w void_ret
119: SCANCONV_PAR:
120: do.w PTS_ary
121: do.w int_val
122: do.w void_ret
123: AA_SCANCONV_PAR:
124: do.w PTS_ary
125: do.w int_val
126: do.w int_val
127: do.w int_val
128: do.w int_val
129: do.w void_ret
130: AA_PAINT_PAR:
131: do.w int_val
132: do.w int_val
133: do.w int_val
134: do.w int_val
135: do.w int_val
136: do.w int_val
137: do.w void_ret
138: TILE_GET_PAR:
139: do.w int_val
140: do.w int_val
141: do.w int_val
142: do.w int_val
143: do.w int_val
144: do.w void_ret
145: TONE_GET_PAR:
146: do.w int_val
147: do.w int_val
148: do.w int_val
149: do.w int_val
150: do.w int_val
151: do.w void_ret
152: WHITEPAPER_PAR:
153: do.w void_ret
154: REVERSE_PAR:
155: do.w void_ret
156: MASKCLEAR_PAR:
157: do.w void_ret
158: MONOTONE_PAR:
159: do.w void_ret
160:
161: * 関数へのポインタ
162: PTR_EXEC:
163: do.l _pts_curve
164: do.l _pts_append
165: do.l _pts_move
166: do.l _pts_oversample
167: do.l _lines
168: do.l _aa_lines
169: do.l _scanconv
170: do.l _aa_scanconv
171: do.l _aa_paint
172: do.l _tile_get
173: do.l _tone_get
174: do.l _whitepaper
175: do.l _reverse
176: do.l _maskclear
177: do.l _monotone
178: .even

```


後処理によるジャギーの除去

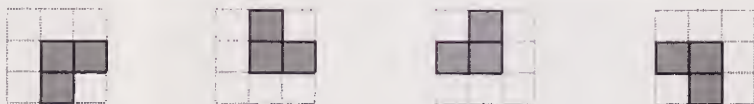
Nakano Shuichi

グラフィック

表 1

```
key 1,"files @@"
key 2,"load @M"
key 3,"auto "
key 4,"list @N"
key 5,"run @M"
key 6,"/*"
key 7,"width "
key 8,"end"
key 9,"func "
key 10,"system"
key 11,"chdir @@"
key 12,"chdrv @@"
key 13,""
key 14,""
key 15,""
key 16,"sa.@@test@M:gbc test||test@M"
key 17,"sa.@@test@M:ed test.bas@M"
key 18,"img_1@Aoad(@A@I@I@I@I@M"
key 19,""
key 20,""
```

图2



チャイルドプロセスを使う。さらに、いちいちチャイルドプロセスを起動してコンパイラにたくさんのオプションを与えるのは面倒なので、コンパイラの起動はバッチファイル、BASICからはファンクションキー1発でコンパイル実行できるようにするとよい。

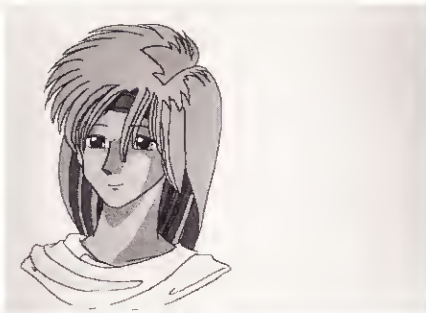
表1のようなファンクションキー設定だとシフト+F6キーで即座にコンパイル実行できる。RUNコマンドの代わりと思えばいい（メモリの少ない人はできません）。

輪郭パターンでの補正

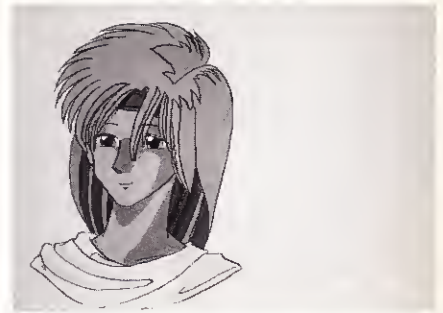
まず2月号で行った局所補間つき画面拡大プログラムを見てみよう。これは256×256ドットの絵を512×512ドットに拡大するものだ。ドットをそのまま大きくすると当然モザイクになる。かといって単純に周辺の色と補間して拡大するとボケボケの絵になる。これを防ぐため、輪郭部分を保護しつつ、全体にぼかしをかけることになった（ただし手抜き処理なので斜め方向は見えない）。

今回のアンチエイリアシング（正確には違うが）でもぼかしを使うことを考えてみよう。絵の輪郭を抽出することは容易だが、そこからベクトルを得ることはちょっと難しいので本格的な処理は私にはできない。

2月号では取り込み画像を対象にしていたため、輪郭保護に重点をおいて明度変化



元画像（協力：高橋哲史）



変換後

の激しい部分ははっきりと、それ以外の部分にぼかしをかけていた。今度はこれとは逆に、明度変化の緩やかな部分は元絵を残し、明度差の激しい部分を選択的にぼかすことになる。しかし、なんでもかんでもぼかすと元絵を大きく損なうので、ぼかさなくてもいい場合を考えよう。

中心が黒でかつ、上下や左右にも黒い点が連続するときはぼかす必要はない（図1）。あまり考えずにアンチエイリアシングをやったよさそうなのは、図2に示されるパターンだ、としよう。

まず、輪郭線部分を取り出し、その周辺の状況（輪郭が連続しているかどうか）を配列に読み込む。ある点の周りには8つの点が存在するので、これをビットごとにchar型配列に入れる。

すると256とおりの場合分けができるので、一気にswitchで最適な処理をすると



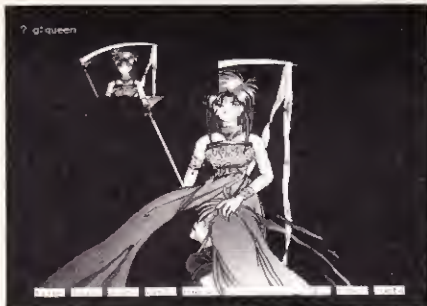
拡大するとこうなる

いうのもいいんだが、ここでは最小限の処理にとどめておく。拡張はご自由に。

中心点が黒かどうかで図2の上下の処理を選択し、ほぼ全ドットに渡って置き換えを実行する。ぼかしは上下左右のドットの色をRGBごとに重みつきで平均することで行っている。点ごとにだぶった処理を行っているがとりあえず気にしない。これでかなりジャギーが減ったはずだ。

リスト1

```
10 /* ----- initialize ----- */
20 screen 1,3,1,1
30 str nam
40 int g_dat(4,2),col,d(4,2),c(4)
50 int blue=0,red=1,green=2,i,q=3333
60 char fl(511,511),fl2(511,511)
70 /* ----- main ----- */
80 input nam
90 pic_load(nam+".pic",0,0)
100 edge():beep
110 jag():beep
120 bokasi()
130 input i
140 end
150 /* ----- */
160 func edge()
170 for y=1 to 510
180   for x=1 to 510
190     c(0)=point(x, y)
200     if c(0)=0 then {
210       fl2(x,y)=1:psat(x,y,1)
220       /* c(1)=point(x+1,y) */:/* 2 エッジ検出部の名残
230       /* c(2)=point(x,y-1) */:/* 4 0 1
240       /* c(3)=point(x,y+1) */:/* 3
250       /* c(4)=point(x-1,y) */:/*
260       /* if c(1)>1 or c(2)>1 or c(3)>1 or c(4)>1 then {
270       /*   fl(x,y)=1
280       /* }
290       /* else fl2(x,y)=0
300     }
310   next
320 endfunc
330 /* ----- */
340 func jag()
350 for y=1 to 510
360   for x=1 to 510
370     col=0
380     /* if fl2(x-1,y-1)=1 then col=col+128
390     /* if fl2(x,y-1)=1 then col=col+64
400     /* if fl2(x+1,y-1)=1 then col=col+32 /* 7 6 5
410     /* if fl2(x-1,y)=1 then col=col+16 /* 4 3
420     /* if fl2(x+1,y)=1 then col=col+8 /* 2 1 0
430     /* if fl2(x-1,y+1)=1 then col=col+4
440     /* if fl2(x,y+1)=1 then col=col+2
450     /* if fl2(x+1,y+1)=1 then col=col+1
460     fl(x,y)=col
470   next
480 next
490 endfunc
500 /* ----- */
510 func bokasi()
520 for y=1 to 510 /* ぼかし処理
530   for x=1 to 510
540     if fl2(x,y)=1 then {
550       if (fl(x,y) and &B10010) >16 then fuz(x,y,0)
560       if (fl(x,y) and &B1010) >8 then fuz(x,y,0)
570       if (fl(x,y) and &B1001000) >64 then fuz(x,y,0)
580       if (fl(x,y) and &B1010000) >64 then fuz(x,y,0)
590     } else {
600       if (fl(x,y) and &B1010000) >64 then fuz(x,y,4)
610       if (fl(x,y) and &B1001000) >64 then fuz(x,y,4)
620       if (fl(x,y) and &B1010) >8 then fuz(x,y,4)
630       if (fl(x,y) and &B10010) >16 then fuz(x,y,4)
640     }
650   }
660 next
670 next
680 endfunc
690 /* ----- */
700 func fuz(x,y,p)
710 c(0)=point(x, y)
720 if c(0)<1 then {
730   c(1)=point(x,y+1)
740   c(2)=point(x+1,y)
750   c(3)=point(x-1,y)
760   c(4)=point(x,y-1)
770   get_rgb()
780   for m=blue to green
790     d(i,m)=(g_dat(0,m)*p+g_dat(1,m)+g_dat(2,m)+g_dat(3,m)+
800     g_dat(4,m))/5*(p+4)
810   next
820   psat(x, y, rgb(d(0,red),d(0,green),d(0,blue)))
830 }
840 endfunc
850 /* ----- */
860 func get_rgb()
870 for m=blue to green
880   for l=0 to 4
890     g_dat(1,m)=(c(1) mod (1 shl(5*m+6)))shr (m*5+1)
900   next
910 next
920 /* ----- */
```

縮小中

ただし、ばかしを直接画面に描いているので、画面処理されたあとのデータを対象に処理が進んでしまう。これはふつうダサイやり方と呼ばれる。スキャンラインごとに処理をすることもできるので、小さなバッファを取って影響がなくなってから書き込むというのが正しいのだろう。多少処理が複雑になることと、モノがばかしだけに周りに影響が出て問題ないんじゃないかという楽観論からこのままにしておいた。

本当は画面分バッファを取って、

```
int gbuff1(511,511)
```

のようにしたかったのだが、こういった配列を2つ取ると多くの人のメモリでは収まらないはずなのであきらめた。

それでも512Kバイト分の配列を取っているので、BASIC.CNFを変更してフリーエリアを広げておいてほしい。あとは、PIC.FNC(1990年6月号)をお持ちの方はそのまま、ない方は“pic_”を“img_”に変更して使えばいい。

なお、PIC.FNCを使ったプログラムをコンパイルする場合、ほかのヘッダファイルなどをいじらない限り、パラメータを省略することはできないので注意しよう。ロード先頭座標やセーブする範囲はその都度指定する。当然コンパイル時にはPICLIB.Aも指定すること。

256ドット縮小

さて、アンチエイリアシングは悪くいえば不十分なドット数をこまかす手法だ。ふつうはオーバーサンプリングといって「たくさんドットがある」つもりで計算しておき、「実は少なかったんだ」といって1ドット

図3



4点の輝度の構成を平均して新しい輝度を得る

に詰め込むときに平均を取ってやる。

X68000の512×512ドットというのは十分なようで実は少ないともいえる。32ビットマシンなら1024×1024以上が標準だろうし、グラフィックのジャギーを見るとこれくらいはほしくなる。しかし、現状のツールでは真真正直な線しか考えてない。しかたないからアンチエイリアシングするわけだが、現状の512×512ドットのモードがすでにオーバーサンプリングされているとみなすとうるだろうか？

実用上必要なのは綺麗な絵であって高い解像度ではない。なにかとかさむ高解像度の絵より256×256ドットの絵が好まれる場合もある。当然、解像度が低いとジャギーが目立つわけだ。256ドット以下のワンポイント的に使われる絵だって綺麗なほうがいいに決まっている。

となると話は簡単。512×512ドットモードで(当たり前のグラフィックツールを使って)描いた絵を縮小してやればいい。手抜きグラフィックツールを使うとドットを間引かれるので、ここではプログラムによって平均化された画像を作ることしよう。

図3のようになった4点をRGB別にして平均し色を決める。小さな画面ならわざわざファイルに書き出したり、大きなバッファを取らなくても画面にそのまま表示できるので結果はリスト2のように単純だ。

ここでは画面の初期化やファイルのロード/セーブを行っていない。BASICで実行しても耐えられない速度ではないということが理由だが、コンパイルして実行したほうがいいに決まっている。必要な人は各自で対応してほしい。また、ファイルのロード時にわざわざinputを使うのも面倒だという場合はコマンドラインから文字を取り込むようにするとよい。Cユーザーズマニュアル参照のこと。

1/4補正つき拡大

なにも画像を小さくしなくても、疑似的にオーバーサンプリングできるようにする手もある。簡単にいえば昔使った4倍拡大アルゴリズムで拡大しておいて、今度はそれをモザイク化して1/4の画像を作り出す、という手だ。拡大時に輪郭補正と周りとの平均化を行うので、不正確ながら高解像度のデータを合成することができよう。

あとは通常のアンチエイリアシングと同様に面積比(といっても4つの平均だが)で色を決定すればいいわけだ。

今回は輪郭色を黒のみに限定して黒のみの補間を行うことにする。それ以外の色ではなにもしない。理由はすぐに縮めるんだからなにもしなくても変わらないからと、黒を残しておけば最初に作ったプログラムをそのまま使ってさらにアンチエイリアシングを図ることもできるからだ。

こうしてできたプログラムがリスト3。画面上の256×256の部分512×512のエリアに拡大する。あらかじめ512×512ドットの絵を1/4ずつに分けてセーブしておいてほしい。

プログラムは同じ画面でもかちあわないように画像の右下から順に処理を進めていく。まず基準点の色を拡大された部分の右隅に打ち、上、左、左上の各ドットの内容から残りの3点の状況決定する。「両方とも黒ならあいだも黒」というのが基本コンセプトだ。

これだと、左斜めは検出するが、逆の斜めは検出できないので逆斜め専用のループも入れてある。

基準点が黒以外ならなにもしないでその色を4点に置く。このあたりは改良の余地があるかもしれない。

輪郭を黒に限定しない場合なら、単に画

リスト2

```
10 int c(3)
20 int blue=0,red=1,green=2
30 int g_dat(3,2),r,g,b
60 for y=0 to 255
70   for x=0 to 255
80     c(0)=point(x*2,y*2)
90     c(1)=point(x*2+1,y*2)
100    c(2)=point(x*2,y*2+1)
120    c(3)=point(x*2+1,y*2+1)
210    for m=blue to green
220      for l=0 to 3
230        g_dat(l,m)=(c(l) mod (1 shl (5*m-6))) shr (m*5-1)
240      next
250    next
260    b=(g_dat(0,blue)+g_dat(1,blue)+g_dat(2,blue)+g_dat(3,blue))%4
270    r=(g_dat(0,red)+g_dat(1,red)+g_dat(2,red)+g_dat(3,red))%4
280    g=(g_dat(0,green)+g_dat(1,green)+g_dat(2,green)+g_dat(3,green))%4
290    pset(x,y,rgb(r,g,b))
300  next
310 next
```


面を1/4ずつに分割して2月号の拡大ルーチンにかけ（自然画でなければ閾値tを多少大きくしたほうがいい）、今月の縮小ルーチンで縮めて4つ並べるだけですむ。

労を惜しまず最高のものを得たいなら、適当に下描きした絵を4分割して拡大修正し、また縮小するという手もある。これならふつうのグラフィックツールを使って処理できる。

2Dグラフィックの今後

もともとは取り込み画像に色をつけようということから始まった。

まずは高橋哲史君が編集室のスキヤナを使って取り込んだ元絵に色をつけようと苦戦している図を想像してもらいたい。ペイントしようとしても途中で止まってしまう。今度はモノクロ2階調で取り込んでペイントしてみる。ちゃんとペイントできるが悲しいくらい絵が粗い。

その場合は、2値化して取り込んだ輪郭線を細くして色を塗り、その上に多値化された綺麗な輪郭線を合成する、という方法で落ち着いた。そして、考えられたのが丹氏の多階調境界対応のペイントルーチンだった。

その後、福原君の手作業によるアンチエイリアシングを見るにつけ、通常のグラフィックツールの限界と可能性を思い知った。確かに境界線を綺麗に処理してやると非常に高画質な絵が得られることはわかった。しかしそれを手作業で行うというのはあまり

に非人間的な作業だろう。これはある程度自動化できそうな処理に思われた。

グラフィックツールはいまだにZ'sSTAFFを最高峰にしたまま進化が止まっている。Z'sSTAFFがよくできたグラフィックツールであることは間違いない。しかし、そろそろもっと凄いものが出てきてもいいんじゃないだろうか。

* * *

さて、なにはともあれ、必要になるのは十分なメモリだ。たとえばSX-WINDOWでまともなアプリケーションが出てきたとすると、あつというまにメモリが足らなくなるだろう。Macintoshと違いSX-WINDOWは複数のアプリケーションを同時実行することを基本に作られているのでメモリはいくらあっても余ることはない。

考えてみれば、多くの初代X68000やACEユーザーは4万円近く払って1Mバイトの増設を行ったわけだ。それが最近では2MバイトのRAMボードが4万円台で買えるようになってきている。X68000を2,3年も使い込んだユーザーなら、そろそろ増設を考えてもよい頃だろう。効果を考えれば決して高い買い物ではない。

特にグラフィック関係はメモリを大量に必要とする場合が多い。メモリさえあれば内部バッファを1600万色分取って表示部だけ65536色にするなどの方法でより高画質なものを作れる。現状の65536色というのは使っていて極端に不足を感じさせる色数ではない。

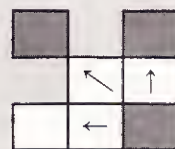
グラフィックツールでグラデーションを

かけたときやレイトレーシングなどを行ったとき以外不足に感じることはないと思う。どちらも表示関係のルーチンをなんとかすれば、65536色でもかなり自然な表示ができるはずだ。うまくやればグラデの縞模様もマッハバンドも出ない。それは今回鈴木氏の256色化や6月号のSXCONVの16色化を見てもわかると思う。

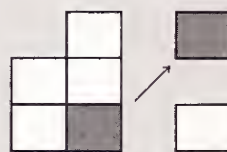
同様に256色モードでも内部で多色処理すればもっと高度なグラフィックツールができるはずだ。しかし、問題は65536色で描いて変換したほうが綺麗だということだろうか。

65536色モードのデータなら扱いやすくほかのモードへの変換も容易(?)だろう。今後の標準はやはりPIC形式の65536(32768)色となるのだろうか？

図4



原点が黒のときは
上、左、斜めを調べる



X軸逆方法から逆斜めを調べる

リスト3

```
10 screen 1,3,1,1
20 str na
30 int i,j,k(3),l,m,n,o,pl,col(3,2),b,r,g,t=8
40 input na
50 pic_load(na+".pic",0,0)
60 for i=0 to 255
70   for j=0 to 255
80     k(0)=point(255-j,255-i)
90     k(1)=point(255-j,255-i-1)
100    k(2)=point(255-j-1,255-i)
110    k(3)=point(255-j-1,255-i-1)
111    pset(511-j*2,511-i*2,k(0))
112    if k(0)=0 then {
130      if k(1)=0 then pset(511-j*2,511-i*2-1,0) else pset(511-j*2,511-i*2-1,k(0))
132      if k(2)=0 then pset(511-j*2-1,511-i*2,0) else pset(511-j*2-1,511-i*2,k(0))
134      if k(3)=0 then pset(511-j*2-1,511-i*2-1,0) else pset(511-j*2-1,511-i*2-1,k(0))
135    } else {
140      pset(511-j*2,511-i*2-1,k(0))
142      pset(511-j*2-1,511-i*2,k(0))
144      pset(511-j*2-1,511-i*2-1,k(0))
149    }
150   next
160   for j=0 to 255
180     k(0)=point(255-j,255-i)
190     /* k(1)=point(255-j,255-i-1)
200     /* k(2)=point(255-j+1,255-i)
210     k(3)=point(255-j+1,255-i-1)
220     if k(0)=0 and k(3)=0 then pset(511-j*2+1,511-i*2-1,0)
230   next
240 next
245 pic_save("ex_"+na,0,0,511,511)
250 input i
260 end
```


色数の補間と量子化

グラフィックデータを変換する

Suzuki Yasuhiro 鈴木 康弘

X 68000にはいくつかの種類の画面モードが存在します。そのなかでも、グラフィックにもっとも適しているのは、やはり512×512ドットの65536色モードでしょう。Z's STAFF PRO-68Kが扱うのも、この画面モードですし、PICなどの圧縮ツールもこの画面モード専用です(最近、ほかの画面にも対応しているAPICというものもあるが)。

ところで、SX-WINDOWが対応しているグラフィックの画面モードは、768×512ドットの16色モードです。ちなみにあふれている、PC-9801などのグラフィックデータは、640×400ドットの16色で描かれています。これらの16色のグラフィックデータと、65536色のデータで決定的に違うことは、16色のデータは、タイリングを用いて中間色を表現しているのに対し、65536色のデータは、タイリングを用いず、中間色はそのままドットの色となっていることです。

今回のプログラムは、これらの16色で描かれたグラフィックデータを、512×512ドットの65536色モードのデータに変換を試みたものです。ただし、そのまま変換すると、タイリングされたまま65536色のデータになってしまい(当然65536色中の16色しか使わない)、全然65536色を使っている気分になりません。

また縦横比を調節すると(640×400を512×512に変換するので、1ドットの大きさが変わってくる)、タイリングパターンが崩れてしまい、元のデータよりも汚くなってしまう。そこで、タイリングで塗ら

れた領域を、なんとかしてそれに対応する色に変換しなければなりません。

逆に色数の多い画面モード用のデータを色数の少ないモード用にコンバートするアルゴリズムは広く知られていますので、それらを使って65536色のデータを256色モードのデータに変換するプログラムも作ってみました。256色モードはグラフィック画面が2枚あり、どうしてもグラフィック画面が1枚では足りないような場合に威力を発揮します。

こっちはほうは、以前Oh!Xで紹介された、オーダーディザ法と、栗野雅彦氏がプリンタのハードコピー用に考え出されたアルゴリズムを応用したものを用いています。また、使われている色数が256色以下の場合、わざわざディザ法を用いるまでもなく256色モードに変換できるので、その処理を行うこともできます。そのほか、画面中でもっともよく使われている256色を抜き出し、それ以外の色をもっとも近い256色で置き換えるというアルゴリズムも発表されていましたが、今回はそれには対応していません(Oh!X1988年2月号参照)。

ちなみに、65536色に変換するとか書いてありますが、実は32768色に変換します(輝度ビットを無視しています)。また、65536色のデータを256色に変換するのではなく、32768色のデータを256色に変換します(輝度ビットのみ異なる色は、同じ色とみなしています)。

コンパイルの方法

プログラムはC言語で書かれています。したがって、XCが必要になるわけですが、XCでコンパイルされたものはとんでもなく処理速度が遅いのです。そこで、Oh!Xの6月号の特別付録にGCCが掲載されているので、できるだけこっちのほうでコンパイルしてください。

GCCでのコンパイル方法は、
gcc T2F.c -O -fstrength-reduce -fo

グラフィックモードの違いを埋める処理に挑戦してみましょう。PC-9801などに描かれた16色のグラフィックデータの色数を増やしてX68000の65536色のデータに変換したり、65536色のデータをできるだけ原画に忠実な256色に変換する際に必要な処理を考えてみます。

```
mit-frame-pointer -liocs -ldos
gcc to256.c -O -fstrength-reduce -fomit-frame-pointer -liocs -ldos
です。ちなみに、XCのほうは、
cc T2F.c -O -Y
cc to256.c -O -Y
です。
```

使い方

まず、16色を32768色に変換するT2F.xですが、あらかじめ、画面を16色モードに設定し(実画面のサイズは1024×1024にしてください)、グラフィックデータを表示しておいてください。T2F.xは、VRAMにあるデータを変換します。そして、グラフィックデータが640×400ならば、

```
T2F -S640
データが512×512ならば、
```

```
T2F -S512
としてください。これでとりあえず変換を開始します。
```

また、タイリングパターンを認識して中間色に変換していくので、認識するタイリングパターンの最大値を指定することができます(省略すると、2ドットになります)。たとえば、640×400ドットのデータで、タイリングパターンの最大値を4ドットにするならば、

```
T2F -S640 -T4
となります。
```

この、タイリングパターンの最大値というのは、最大何ドットでタイリングされているか、というものです。たとえば、

黒白白黒白白……

というタイリングがある場合には、最大値に3以上を設定しなければ、これはタイリングとみなされず、そのまま残ってしまいます。

この値をむやみに大きくすると、タイリングでないところまでタイリングとみなしてしまい、変なところが1色で塗られてしまいますから、注意してください。



オーダーディザ法による変換

次に、32768色のデータを256色に変換する、to256.xですが、これもグラフィックを表示させてからプログラムを実行させてください。

使い方は、スイッチに、オーダードディザ法で変換する場合には“-D”，桑野式アルゴリズムで変換する場合には“-K”，色数を数えて、256色以下の絵をそのまま変換する場合には“-C”をつけ加えて起動してください。

オーダードディザ法で変換する場合には、閾値を指定することができます。たとえば、閾値に60を設定したいのなら、

to256-D60

のように、“-D”に続けて閾値を書きます。省略すると40が設定されます。この値はグラフィックの内容によって最適な値が変わるので、いろいろ試してください。

タイリングについて

16色モードのグラフィックは、ほとんどがタイリングという手法を用いています。このタイリングというのは、たとえば、赤と青のドットを交互に並べていくと、遠目には紫色に見えてしまう、というものです。これを用いると、16色しか出ないはずなのに、それ以上の色を表現することができるのです。

●16色→32768色

まず、タイリングされているグラフィックデータをよ〜く見てみますと、タイリングが施されている部分はかなりの規則性があることがわかります。つまり、ある決まったドットの並びが横にず〜っと並んでいるのです。色が変わる部分というのは、その決まったドットの並びに合わなくなる部分なのです。

さて、この変換の大まかなアルゴリズムを説明します。

まず、最初にタイリングパターンを横方向に比較していき、そのタイリングパターンが崩れたドットに、フラグを立てて覚えておきます。この処理を全画面に行うと、タイリングパターンが変化した部分（要するに、遠くから見たときの、色に変化する部分→輪郭）にフラグが立つことになります。

あとは、このフラグとフラグのあいだを、その中のタイリングパターンで色で塗ってあげばよいのです。

この変換の核となるタイリングパターンが変化した部分の認識ですが、次の手順で行っています。

- 1) あるドットから右にnドット分を配列変数に格納する
- 2) さらに、その右nドットが、配列変数に格納した色と同じかどうかを調べる
- 3) 同じならば、そこからタイリングパターンが続いていることになる
- 4) 違うのなら、nにn-1を設定して、もう1回調べなおす（1に飛ぶ）
- 5) nが1になってしまったら、そのドットからはタイリングは始まっていない。したがって、そこにフラグを立てて、1ドット右に移動し、新たに調べ始める（1に飛ぶ）

これで、タイリングパターンが続いているかどうかわかります。これがわかったら、次はどこまで続いているかです。これは、次々に配列の内容と実際のドットとを比べていき、それらが異なるところまでとなります。

例を出してみると、

座標 0 1 2 3 4 5 6 7 8 9

色 赤 青 赤 青 赤 青 赤 青 黒 黄

10 11 12 13 14

赤 黄 赤 黄 赤 ……

というドットの並びの場合、まず、先頭の赤青を配列変数に入れます（タイリングの最大値が2の場合）。次に、座標2からの2ドットが、配列変数に入っているものと同じかどうかを調べます。この場合は同じですので、これで赤青というタイリングがあると認識します。

次に、2ドット右に進んで、配列の内容と同じかどうかを調べます。同じですので、さらに2ドット進んで調べます。

どんどん右に調べていくと、座標8の部分で、配列変数の内容と食い違う色が出てきます。そこで、この座標8のドットにフラグを立てます。

さらに、そこから2ドットを配列に入れます。この場合、黒黄が入ります。ところがいきなり次の2ドットと色が異なるため、この黒から始まるのはタイリングではないとみなし、座標9の黄色にフラグを立てます。

そして、次の2ドット（黄赤）を配列に入れ、再び調べ始めます。

この場合、最初に2ドットを配列に入れて調べましたが、この数値は変更することができ、たとえば3ドットにしてあると、まず3ドットを配列に入れ、そのパターンが続かなければ2ドットにして調べるようになっていきます。

あと、タイリングパターンから色を決める方法ですが、これは単純に、各タイリン



桑野式アルゴリズムによる変換

グパターンのドットのRGB成分の平均を出し、そのRGBの平均によって作られる色になります。

2ドットのタイリングでしたら、

求める色 = (色1 + 色2) ÷ 2

になります。

●640×400→512×512

実は、今回のプログラムでは、あまりにも色の決定の部分のアルゴリズムの部分に時間をさいてしまい、縦横比の変換はかなりいい加減になっています。したがって、640×400のグラフィックを変換しても、そんなに綺麗にはなりません。

具体的にどうやっているかをばらしますと、まず640×400の32768色が記憶できるバッファを取り、タイリングパターンを調べて色を塗るところまでは、そのバッファに対して処理を行います。その後、画面に512×512で表示する段階になったら、1ドットずつ、対応するドットを調べて、それを画面に表示しているのです。したがって、横方向はとところどころドットが抜けて表示され、縦方向はとところどころ同じドットが2ドット続きます。

この、抜けたり、2ドット続いたりするのが輪郭の部分だったりすると、輪郭が抜けたり、太くなったりしてしまいます。試しに、640×400のグラフィックでも、512×512で変換してみてください。とりあえず、輪郭は綺麗に変換されると思います。

縦横比を調節すると、輪郭が太くなってしまうというのも欠点ですが、まだあります。

このプログラムでは、横方向しかタイリングを調べていないので、たとえば、

赤 白

白 白

という、2×2のタイリングパターンには無力です。

実は、このプログラムの最初のバージョンでは、縦方向も調べていたのですが、横方向で調べた輪郭を認識しない部分が出てくるなど、いろいろ問題点が多かったのだ

す。そこで、これなら縦方向を無視したほうがいだろうと思ひ、横方向のみとなつたわけです。

●32768色→256色

こちらのプログラムでは、オーダードディザ法と桑野氏のアルゴリズムのどちらかで変換できます。それぞれの詳しい原理などは、以前のOh!Xに載っています。オーダードディザ法は、1988年9月号で丹氏が、桑野式アルゴリズムは、1988年11月号で桑野氏が説明しています(1990年6月号にも掲載されている)から、そちらも参照してください。

オーダードディザ法については電腦倶楽部に最近掲載されたものとアルゴリズムから参考文献まで同じですので、同様の実行結果になるようです。

桑野式(もしかしたら、これが誤差拡散法なのだろうか?)は画面の情報量を減らさずに色変換をする優れたアルゴリズムです。たいていの場合、ディザ法を使うよりも自然

な仕上がりになるようです。

RBGという分け方で見ると、256色というのは半端な値なのですが、ここでは6月号のSXCONV(これは65536色を16色に変換する)と同様な考え方にに基づき、

GGRRRBBB

と、もっとも輝度の低そうな青成分を2ビット、ほかを3ビットで処理することによって自然な色に変換しています。ちなみにX68000標準のパレット設定では、

GRRRRBBB

のように、緑が2ビットで処理されています。

色数が256色以下のグラフィックについては、パレットを変更することによりそのままの画像で256色モードに変換できます。256色しか使われていないグラフィックは少ないように思えるかもしれませんが、レイトレイ取り込みなどを除く、人が描いたようなグラフィックでしたら、たいていの場合256色以下しか使われていません。

おわりに

最初は画面全体にボカシをかけて、色が急に变化する部分を見つけ出し、そこを輪郭として色を塗っていく、という路線で作っていましたが、どうもうまく輪郭が認識できませんでした。

256色に変換するというのも、使われている色数が256色以上の場合、似た色を同じパレットに割り当てる、という路線で攻めていましたが、いまいち実行速度が遅くなります。なんとかして高速化を図ろうとしたのですが、いつのまにかオーダードディザ法と桑野式アルゴリズムに落ち着いてしまいました。

今回はええいくそ、という掛け声とともに削除されたファイルが数知れず(その直後に、しまった、という掛け声とともに復活されたファイルも数知れず)、なのでした。

リスト1

```
===== T2F.c =====
1: /*
2: /*      16色→32768色 コンバータ version 1.20
3: /*
4: /*      by Yasuhiro Suzuki
5: /*
6: /*
7:
8: #include <stdio.h>
9: #include <stdlib.h>
10: #include <doslib.h>
11: #include <localib.h>
12:
13: #define ushort  unsigned short
14: #define uint      unsigned int
15:
16:
17: /*****
18: /* グローバル変数の宣言 */
19: /*****/
20: ushort  *buf;
21: int  xsize = 640;
22: int  ysize = 400;
23: int  tmax = 2;
24: int  pg[16];
25: int  pr[16];
26: int  pb[16];
27:
28: /*****/
29: /* バッファを初期化する */
30: /*****/
31: void bclr()
32: {
33:     ushort *p = buf;
34:     int i;
35:     for( i=xsize*ysize; i>0; i-- ){
36:         *(p++) = 1;
37:     }
38: }
39:
40: /*****/
41: /* パレットをRGBに分解して保存 */
42: /*****/
43: void ptrns()
44: {
45:     ushort *p = (ushort *)0xEB2000;
46:     int i, c;
47:     for( i=0; i<16; i++){
48:         c = *(p++);
49:         pg[i] = (c >> 11) & 0x1F;
50:         pr[i] = (c >> 6) & 0x1F;
51:         pb[i] = (c >> 1) & 0x1F;
52:     }
53: }
54:
55: /*****/
56: /* コマンドオプションを調べる */
57: /*****/
58: int chksw( ac, av )
59: {
60:     int ac;
61:     char *av[];
62:     {
63:         int i, c;
64:         for( i=1; i<ac; i++){
65:             if( ( av[i][0] == '-' ) || ( av[i][0] == '/' ) ){

```

```

66:                 c = av[i][1] & 0x20;
67:                 if( c == 's' ){
68:                     xsize = atoi( av[i+1] );
69:                     if( xsize == 0 ){
70:                         ysize = 512;
71:                     }
72:                     else if( xsize == 640 ){
73:                         ysize = 400;
74:                     }
75:                     return( 1 );
76:                 }
77:                 else if( c == 't' ){
78:                     tmax = atoi( av[i+1] );
79:                     if( tmax == 0 ){
80:                         return( 1 );
81:                     }
82:                 }
83:                 else{
84:                     return( 1 );
85:                 }
86:             }
87:         }
88:         return( 0 );
89:     }
90: }
91:
92: /*****/
93: /* タイリングのドット数を調べる */
94: /*****/
95: int tlen( vp, max )
96: {
97:     ushort *vp;
98:     int max;
99:     {
100:         ushort *p, ti[256];
101:         int i, j, r;
102:         if( max > tmax ){
103:             max = tmax;
104:         }
105:         for( i=max; i>1; i-- ){
106:             p = vp;
107:             for( j=0; j<i; j++){
108:                 ti[j] = *(p++);
109:             }
110:             r = 1;
111:             for( j=0; j<i; j++){
112:                 if( ti[j] != *(p++)){
113:                     r = 0;
114:                     break;
115:                 }
116:             }
117:             if( r ){
118:                 return( i );
119:             }
120:         }
121:         return( 0 );
122:     }
123: }
124:
125: /*****/
126: /* 境界色を書き込む */
127: /*****/
128: void tset( x, y, ti, n )
129: {
130:     int x, y, n;
131:     ushort *ti;
132:     int i;

```



```

141:  uint g, r, b, c;
142:
143:  g = r = b = 0;
144:  for( i=0; i<n; i++, ti++ ){
145:      g += pg[*ti];
146:      r += pr[*ti];
147:      b += pb[*ti];
148:  }
149:
150:  g /= n;
151:  r /= n;
152:  b /= n;
153:  c = ( g << 11 ) | ( r << 6 ) | ( b << 1 );
154:
155:  *( buf + (int)( x + y * xszsize ) ) = c;
156: }
157:
158: /*****
159:  * タイリングを調べる */
160: /*****
161: void tilex()
162: {
163:     ushort ti[256];
164:     ushort *vp, *vvp;
165:     int x, y, t, i;
166:
167:     vvp = (ushort *)0xC00000;
168:     for( y=0; y<ysize; y++ ){
169:         t = 0;
170:         vp = vvp;
171:         vvp += 1024;
172:         for( x=0; x<xsize; x ){
173:             if( t == 0 ){
174:                 if( t = tlen( vp, ( xszsize - x ) / 2 ) ){
175:                     for( i=0; i<t; i++){
176:                         ti[i] = *(vp+i);
177:                     }
178:                     tset( x, y, ti, t );
179:                     x += t;
180:                 }
181:                 else{
182:                     ti[0] = *(vp++);
183:                     tset( x++, y, ti, 1 );
184:                 }
185:             }
186:             else{
187:                 if( t > ( xszsize - x ) ){
188:                     t = xszsize - x;
189:                     for( i=0; i<t; i++, x++, vp++ ){
190:                         if( ti[i] != *vp ){
191:                             t = 0;
192:                             break;
193:                         }
194:                     }
195:                 }
196:             }
197:         }
198:     }
199: }
200:
201: /*****
202:  * 境界色で塗り潰す */
203: /*****
204: void fullx()

```

```

205: [
206:     ushort *bp, *vp, c;
207:     int x, y, xx, yy;
208:
209:     bp = buf;
210:     for( y=0; y<ysize; y++ ){
211:         c = *bp;
212:         for( x=0; x<xsize; x++ ){
213:             if( *bp != 1 ){
214:                 c = *(bp++);
215:             }
216:             else{
217:                 *(bp++) = c;
218:             }
219:         }
220:     }
221:
222:     vp = (ushort *)0xC00000;
223:     for( y=0; y<ysize; y++ ){
224:         yy = ( y * ysize ) / 512 * xszsize;
225:         for( x=0; x<512; x++ ){
226:             xx = ( x * xszsize ) / 512;
227:             *(vp++) = *( buf + (int)( xx + yy ) );
228:         }
229:     }
230: }
231:
232: /*****
233:  * メインルーチン */
234: /*****
235: int main( ac, av )
236: int ac;
237: char *av[];
238: {
239:     puts( "TILE to FULL ver1.20 by Yasuhiro Suzuki" );
240:
241:     if( chksw( ac, av ) ){
242:         puts( "[ 使用法 ] T2F [ <スイッチ> ] ..." );
243:         puts( "%t-S640x400x400ドットの絵を変換する。" );
244:         puts( "%t-S512x512x512ドットの絵を変換する。" );
245:         puts( "%t-TnYt識別するタイリングパターンのドット数の最大値" );
246:         return( 1 );
247:     }
248:
249:     if( ( buf = (ushort *)MALLOC( xszsize * ysize * 2 ) ) != (ushort *)0
x80000000 ){
250:         puts( "メモリが足りません。" );
251:         return( 1 );
252:     }
253:
254:     SUPER(0); /* スーパーバイザモードになる */
255:
256:     bclr(); /* バッファを初期化する */
257:     ptrns(); /* パレットを保存 */
258:
259:     tilex(); /* タイリングの変化点を調出する */
260:
261:     CRTMOD( 12 ); /* 画面を初期化する */
262:     G_CLR_ON();
263:
264:     fullx(); /* V R A M に表示する */
265:
266:     return( 0 );
267: }

```

リスト2

```

===== to256.c =====
1: /*
2:  * 6 5 5 3 6 色 → 2 5 6 色 コンバータ version 2.13
3:  *
4:  *
5:  * by Yasuhiro Suzuki
6:  *
7:
8:  * #include <stdio.h>
9:  * #include <stdlib.h>
10:  * #include <locale.h>
11:  * #include <doslib.h>
12:
13:  * #define uchar unsigned char
14:  * #define ushort unsigned short
15:
16:  * #define VRAM (ushort *)0xC00000
17:
18:
19:  * ****
20:  * * グローバル変数の宣言 */
21:  * ****
22:  uchar *buf;
23:  ushort pcnv[32768]; /* しきい値 */
24:  int dn;
25:  int bg[2][512];
26:  int br[2][512];
27:  int bb[2][512];
28:  int mat[8][8] = { /* デザ法で使う行列 */
29:      0, 32, 8, 40, 2, 34, 10, 42,
30:      48, 16, 56, 24, 50, 18, 58, 26,
31:      12, 44, 4, 36, 14, 46, 6, 38,
32:      60, 28, 52, 20, 62, 30, 54, 22,
33:      3, 35, 11, 43, 1, 33, 9, 41,
34:      51, 19, 59, 27, 49, 17, 57, 25,
35:      15, 47, 7, 39, 13, 45, 5, 37,
36:      63, 31, 55, 23, 61, 29, 53, 21
37:  };
38:
39:  * ****
40:  * * パレット */
41:  * ****
42:  #define IM (256*31)
43:
44:  int rrr[7] = { IM/7, IM*2/7, IM*3/7, IM*4/7, IM*5/7, IM*6/7,
IM*7/7 };
45:  int ggg[7] = { IM/7, IM*2/7, IM*3/7, IM*4/7, IM*5/7, IM*6/7,
IM*7/7 };
46:  int bbb[3] = { IM/3, IM*2/3, IM*3/3 };
47:
48:  unsigned short rgb[8][8][4];
49:
50:  #define RMAX 31

```

```

51: #define GMAX 31
52: #define BMAX 31
53:
54: #define PALRGB(R,G,B) { ((G)*GMAX/7)&31<<11 | (((R)*RMAX/7)
&31)<<6 | (((B)*BMAX/3)&31)<<1 }
55:
56: /*****
57:  * 色数を数える */
58: /*****
59: int count()
60: {
61:     ushort *vp, c;
62:     int i, k;
63:
64:     for( i=0; i<32768; i++ ){
65:         pcnv[i] = 0;
66:     }
67:
68:     vp = VRAM;
69:     k = 1;
70:     for( i=512*512; i>0; i-- ){
71:         c = *(vp++) >> 1;
72:         if( pcnv[c] == 0 ){
73:             pcnv[c] = k++;
74:         }
75:     }
76:
77:     return( k - 1 );
78: }
79:
80: /*****
81:  * V R A M の内容を256色に変換して格納 */
82: /*****
83: void trns()
84: {
85:     uchar *bp;
86:     ushort *vp;
87:     int i;
88:
89:     bp = buf;
90:     vp = VRAM;
91:     for( i=512*512; i>0; i-- ){
92:         *(bp++) = pcnv[ *(vp++) >> 1 ] - 1;
93:     }
94: }
95:
96: /*****
97:  * オーダーディザで色を変換する */
98: /*****
99: void dither()
100: {
101:     uchar *bp;
102:     ushort *vp, c;

```



```

103: int g, r, b, d;
104: int x, y;
105:
106: bp = buf;
107: vp = VRAM;
108: for( y=0; y<512; y++ ){
109:     for( x=0; x<512; x++ ){
110:         c = *(vp++);
111:         g = ((c >> 11) & 0x1F) << 3;
112:         r = ((c >> 6) & 0x1F) << 3;
113:         b = ((c >> 1) & 0x1F) << 3;
114:         d = mat[ y & 0x07 ][ x & 0x07 ];
115:         g = ( g + d ) / dn / 2;
116:         r = ( r + d ) / dn;
117:         b = ( b + d ) / dn;
118:         if( g > 3 ){
119:             g = 3;
120:         }
121:         if( r > 7 ){
122:             r = 7;
123:         }
124:         if( b > 7 ){
125:             b = 7;
126:         }
127:         *(bp++) = ( g << 6 ) | ( r << 3 ) | b;
128:     }
129: }
130:
131:
132: /* 森野式アルゴリズムで変換 */
133: /* 森野式アルゴリズムで変換 */
134: void kuwano()
135: {
136:     uchar *bp;
137:     ushort *vp, c;
138:     int x, y, lc, lb;
139:     unsigned int cg, cr, cb, dg, dr, db;
140:     int i;
141:
142:     bp = buf;
143:     vp = VRAM;
144:     for( x=0; x<512; x++ ){
145:         bg[0][x] = br[0][x] = bb[0][x] = 0;
146:     }
147:     for( y=0; y<512; y++ ){
148:         lc = y & 1;
149:         lb = ( y + 1 ) & 1;
150:         for( x=0; x<512; x++ ){
151:             bg[1b][x] = br[1b][x] = bb[1b][x] = 0;
152:         }
153:         cg = cr = cb = 0;
154:         for( x=0; x<512; x++ ){
155:             c = *vp;
156:             cg += ((c >> 11) & 0x1F) * 256 + bg[1c][x];
157:             cr += ((c >> 6) & 0x1F) * 256 + br[1c][x];
158:             cb += ((c >> 1) & 0x1F) * 256 + bb[1c][x];
159:
160:             dg=dr=db=0;
161:             for( i=6; i>=0; i-- ){
162:                 if( cg >= ggg[i] ){
163:                     dg = i+1;
164:                     cg -= ggg[i];
165:                     break;
166:                 }
167:             }
168:             for( i=6; i>=0; i-- ){
169:                 if( cr >= rrr[i] ){
170:                     dr = i+1;
171:                     cr -= rrr[i];
172:                     break;
173:                 }
174:             }
175:             for( i=2; i>=0; i-- ){
176:                 if( cb >= bbb[i] ){
177:                     db = i+1;
178:                     cb -= bbb[i];
179:                     break;
180:                 }
181:             }
182:             *(bp++) = ( dg << 5 ) | ( dr << 2 ) | db;
183:             *(vp++) = PALRGB( dr, dg, db );
184:
185:             bg[1b][x] += cg/8;
186:             br[1b][x] += cr/8;
187:             bb[1b][x] += cb/8;
188:             bg[1b][ ( x > 0 ) ? ( x - 1 ) : ( x ) ] += (cg/4);
189:             br[1b][ ( x > 0 ) ? ( x - 1 ) : ( x ) ] += (cr/4);
190:             bb[1b][ ( x > 0 ) ? ( x - 1 ) : ( x ) ] += (cb/4);
191:             bg[1b][ ( x < 511 ) ? ( x + 1 ) : ( x ) ] += cg/8;
192:             br[1b][ ( x < 511 ) ? ( x + 1 ) : ( x ) ] += cr/8;
193:             bb[1b][ ( x < 511 ) ? ( x + 1 ) : ( x ) ] += cb/8;
194:
195:             cg/=2;
196:             cr/=2;
197:             cb/=2;
198:         }
199:     }
200: }
201:
202:
203: /* パレットを設定 */
204: /* パレットを設定 */
205: void setpal()
206: {
207:     ushort *pp, c;
208:     int i;
209:
210:     pp = (ushort *)0xE82000;
211:     for( i=0; i<32768; i++ ){
212:         if( ( c = pcnv[i] ) != 0 ){
213:             *(pp + c - 1) = i << 1;
214:         }
215:     }
216: }
217:
218:
219: /* バッファの内容をVRAMに転送 */
220: /* バッファの内容をVRAMに転送 */
221: void prt()
222: {
223:     uchar *bp;
224:     ushort *vp;

```

```

226: int i;
227:
228: bp = buf;
229: vp = VRAM;
230: for( i=512*512; i>0; i-- ){
231:     *(vp++) = (ushort) *bp--;
232: }
233:
234:
235: /* 画面モード初期化 */
236: /* 画面モード初期化 */
237: void ginit0()
238: {
239:     CRTMOD( 8 );
240:     G_CLR_ON();
241: }
242:
243:
244: void ginit1()
245: {
246:     int r, g, b;
247:
248:     CRTMOD( 8 );
249:     G_CLR_ON();
250:
251:     for( g=0; g<8; g++ ){
252:         for( r=0; r<8; r++ ){
253:             for( b=0; b<4; b++ ){
254:                 rgb[r][g][b] = g<<13 | r<<8 | b<<4;
255:                 GPALET( g<<5 | r<<2, PALRGB( r, g, b ) );
256:             }
257:         }
258:     }
259:     return;
260: }
261:
262: /* コマンドオプションをチェック */
263: /* コマンドオプションをチェック */
264: int chksw( ac, av )
265: {
266:     int ac;
267:     char *av[];
268:
269:     int c, i, r=0;
270:
271:     if( ac < 2 ){
272:         return( 0 );
273:     }
274:
275:     for( i=1; i<ac; i++ ){
276:         if( av[i][0] != '-' ){
277:             return( 0 );
278:         }
279:         c = av[i][1] & 0x20;
280:         switch( c ){
281:             case 'd':
282:                 r |= 0x01;
283:                 dn = atoi( &av[i][2] );
284:                 if( dn >= 0 ){
285:                     dn = 40;
286:                 }
287:                 break;
288:             case 'k':
289:                 r |= 0x02;
290:                 break;
291:             case 'c':
292:                 r |= 0x04;
293:                 break;
294:             default:
295:                 return( 0 );
296:         }
297:     }
298:
299:     if( (( r & 0x03 ) == 0x03 ) || ( r & 0x03 ) == 0x00 ){
300:         return( 0 );
301:     }
302:     return( r );
303: }
304:
305: /* メインルーチン */
306: /* メインルーチン */
307: int main( ac, av )
308: {
309:     int ac;
310:     char *av[];
311:
312:     int m;
313:
314:     puts( "65536 to 256 ver2.13 by Yasuhiro Suzuki" );
315:     if( ( buf = (uchar *)MALLOC( 512 * 512 ) ) != (uchar *)0x80000000 ){
316:         puts( "メモリが足りません。" );
317:         return( 1 );
318:     }
319:
320:     if( !( m = chksw( ac, av ) ) ){
321:         puts( "[使用法] to256 <スイッチ>*1;" );
322:         puts( "/*-Dn*/ オードディ法で変換を行う (nはしきい値)" );
323:     }
324:
325:     puts( "/*-K*/ 森野式変換を行う" );
326:     puts( "/*-C*/ 色数を調べて変換する" );
327:     return( 0 );
328: }
329:
330: SUPER(0);
331:
332: if( ( m & 0x04 ) && ( count() <= 256 ) ){
333:     trns();
334:     ginit0();
335:     setpal();
336: }
337:
338: else if( m & 0x01 ){
339:     dither();
340:     ginit0();
341: }
342:
343: else{
344:     kuwano();
345:     ginit1();
346: }
347:
348: prt();
349:
350: return( 0 );
351: }

```


4096色→8色変換

Zの画像をX1で

Kameda Masahiko 亀田 雅彦

なぜ、8色なの?

今月は大盤振る舞いなのです。まさに「もってけどろぼう!」の世界といえるでしょう。なぜかという、この特集とKAME-DOS連載の豪華2本立てだからです。しかも、それらが見事に調和を保ちながらダブル進行していくという華麗さ、名づけて「シンクロ原稿」です。「ライターがX1関係で荒稼ぎをしようとしてる」とか、「1本のプログラムを使い回してるだけだ」という噂の真偽はさておき、特集とは名ばかり、KAME-DOS関係の話が割り込んでくるので悪しからず。

しかしながら、グラフィック特集である以上グラフィックにも力をいれなければなりません。そこで今回は「Zの4096色画像を8色に変換してみよう」ということになりました。ここでふと思ひ浮かんでくるのは、6月号のSX-WINDOWのグラフィックについて。パラパラとめくってみると、そのものずばり載っているじゃあないですか。しかもその6月号ですら、1988年11月号の引用なのだから、私は「引用の引用」をするという、神をも恐れぬワザにでようというわけです。でも楽しいことはいいことなので、そのまま採用させてもらいました(実際の実行結果も良好でした)。

それじゃ8色に変換してうれしいこと。

●メモリが節約できる

96K(4096色フル)だと2Dディスクで3枚ちょっと。2HD(アクセスが遅い)なら10枚くらいで、結構邪魔です。ディスクアクセス側の問題もありますが、容量はロード時間にも影響を与えます。

●互換性が出てくる

4096色というのはあまりメジャーな数字ではないですが、8,16色あたりはMS-DOSの世界では常識です。もちろんX1のVRAMデータ形式のままで互換性はありませんが、変換自体は簡単にできそうなので挑戦してみるのも面白そうです。

●プリンタとの相性がいい

実はこれが一番身近な問題だと思います。Zではアナログ画像取り込みが標準で装備されながら、あまり活用されないのはグラフィックの扱いにくさが原因でしょう。カラーイメージボードは8色でありながら、そのデータの少なさがよいほうへ働いています。Z標準のアナログ画像を精一杯有効に活用していきたいとすれば、8色に落としてプリンタへの出力を容易にするのが効果的です。ひょっとすると、安価なスキャナとしての価値をZに見出せるかもしれません。

もちろん、8色にして悪いことは原画の情報がRGBの各色について1/4ずつになることです。これをなるべく緩和しようとするのが前述のアルゴリズムです。

プログラムは?

画像変換のためのものと画面ローダ、画面セーバの3本を用意しました。ローダとセーバに関しては、連載のKAME-DOSの外部コマンドとしても使えるようになっていきます(そっちがメインだったりして)。もともとKAME-DOSのほうで外部コマンドの許容範囲が広いので、画像変換プログラムもコマンドとすることができます(あまり意味はありません)。

画像変換プログラムは人のアルゴリズムを使っているのであまり自慢できたものじゃありませんが、ローダとセーバのスピードに関しては自信を持っています。KAME-DOSの実力をいかに発揮させて、理論的な最高速に達しました。画面全体を一度にロードしたりセーブしたりしかできませんが、そのスピードは一度見てもらえればわかります。

こんなスピードを競うようなプログラムは最近では見かけませんが、8ビット全盛の頃はよくはやったものです(特にグラフィック命令)。自分で書いててなつかしくなりました。

入力方法

●画像変換プログラム(リスト1&リスト2)

X1turboZでなおかつZ-BASIC専用プログラムです(必ずしもKAME-DOSは必要ありません)。Z-BASICからリスト1を入力したら、ファイル名はとりあえず「CCHANGE.X1」としてセーブしておいてください。次に、CLEAR &HC000を実行して、リスト2をなんらかのマシン語入力ツールから打ち込んでください。間違いがなければSAVEM「CCHANGE.OBJ」,&HC000,&HC1A7としてセーブします。使うときは両方必要になるので、2つは同一ディレクトリ上においてください。

●画面ローダ

●画面セーバ

X1全シリーズで使うことができます。ただし、6月号から今月にかけて連載しているKAME-DOSシステムが必要になります。具体的には、「INTEGRAL.X」「COMMAND.X1」「FDC.OBJ」の3つのプログラムと、ノーマルX1には7月号のプログラムも必要です。まだ持っていない方は、バックナンバーなどからぜひ入手してください。

「CZ-8FB01,turboBASIC,Z-BASIC」のうちKAME-DOSのあるBASICで、今月の92ページから連載に載っているリストを入力します。「COMMAND.X1」と同一ディレクトリ上にセーブしてください。変数名の間違いがあったりすると、ディスクを破壊しかねないので慎重にチェックしてください。

また、入力上の注意は今月号の連載の「外部コマンド」の入力法のところをよく読んで必ず守るようにしてください。ファイル名はそれぞれ「GLOAD.X1」「GSAVE.X1」とします。

●まとめ

1: Z-BASIC&KAME-DOS

「COMMAND.X1」「CCHANGE.X1」
「CCHANGE.OBJ」「GLOAD.X1」
「GSAVE.X1」を同一ディレクトリ上にお
いてください。

2:Z-BASICのみの方

「CCHANGE.X1」「CCHANGE.OBJ」
を同一ディレクトリ上においてください。

3:KAME-DOSのみの方

「COMMAND.X1」「GLOAD.X1」
「GSAVE.X1」を同一ディレクトリ上にお
いてください。

使い方

●CCHANGE.X1 (KAME-DOSなし)

あらかじめグラフィックを表示させてお
いてCCHANGE.X1を起動します。メニュー
画面になるので、1を押すと全画面に対
して(少し時間がかかりますが)4096色か
ら8色へ変換します。それが終わると、キ
ー入力待ちになって、入力するとメニュー
へ戻ります。メニューの2,3は使えませ
ん。4で終了です。

スペースキーでグラフィックのON/

OFFができます。

●CCHANGE.X1 (KAME-DOSあり)

KAME-DOSのコマンドライン([X:/])
から「CCHANGE」として起動します。
GLOAD.X1,GSAVE.X1があればメニュー
の2,3が使えます。それぞれ選択する
と、ファイル名の入力になるので、ドライ
ブ名を含めてフルパスで指定してください。
リターンキーだけを押せば、メニューに戻
ります。その後の操作はGLOAD,GSAVE
と同じになります。

メニューからグラフィックをロードする
こともできますが、あらかじめロードして
おきたいこともあります。そういうときは、
グラフィックをロードして、「INTEGRAL.
X」の中のグラフィックを消すような命令
を削ってから、KAME-DOSを起動してく
ださい。

●GLOAD.X1

96Kバイトあるいは64Kバイト(自動的
に判断する)のグラフィックファイルをロ
ードします。ファイルの拡張子によって画
面モードを自動変更するので注意してく
ださい(図1)。

KAME-DOSのコマンドラインから
「GLOAD ファイル名」として起動しま
す。エラーがなければ、グラフィックを表
示してキー入力待ちになるので、キーを押
すと親プロセスへもどります。エラーがあ
ればメッセージを表示して実行を中止しま
す。ロードしている最中は少しキャラクタ
画面が乱れますが、それが正常なので心配
いりません。ノーマルX1では96Kファイ
ルはロードできません。

Z-BASICには標準でVLOAD,VSAVE

INTEGRALXを書き換える

次に挙げる命令を、自分のINTEGRALXで削
除してください。ただし、これはグラフィッ
クをあらかじめロードしておいたときのみの
処置なので、通常はいつものINTEGRALXを使
ってください。

1040行のWIDTH・1050行のCLS 4

1200行のINIT

また、INTEGRALXやCOMMAND.X1にある
SCREEN命令はグラフィック画面を見えなく
するものなので、必要に応じて入れておい
てください。Z-BASICを使う場合は、OPTION
SCREEN 4をOPTIONSCREEN 5に換えておき
ましょう。

リスト1

```
1000 'CCHANGE.X1 Ver 1.0 By Kameda
1010 '
1020 OPTIONSCREEN 4:WIDTH 40,25,0,1:OPTIONSCREEN 5:INIT:DEFINT a-z
1030 DEFUSR0=m_tranr
1040 '
1050 CLEAR &HC000:LOADM "CCHANGE.OBJ"
1060 '-----{ MAIN ROUTINE }-----
1070 '
1080 SCREEN:CLS:fes(1)="":GOSUB "menu":CLS
1090 ON a GOTO 1109,"load","save",1160:IF a$=CHR$(27) GOTO 1160
1100 '
1110 KLIST 0:CONSOLE 0,25
1120 MEM$(&HC007,8)=MKIS(0)+MKIS(320)+MKIS(0)+MKIS(200)
1130 OPTIONSCREEN 4:INIT:CFLASH 1:PRINT "Wait a moment.":CFLASH 0
1140 CALL &HC000
1150 GOSUB "ending":GOTO 1080
1160 '
1170 CLS:IF proces=0 THEN END
1180 CLEAR &HD000:proces=proces-1:CHAIN proces$(proces)
1190 '-----{ LOAD }-----
1200 '
1210 LABEL "load"
1220 IF proces=0 THEN 1080
1230 LOCATE 7,7:PRINT "*** GRAPHIC LOAD ***"
1240 LOCATE 11,10:COLOR 1:PRINT "[RETURN]: MENU"
1250 LOCATE 5,13:COLOR 6:PRINT "FILE-NAME":COLOR 7:INPUT "",fes(1)
1260 IF fes(1)=" THEN 1080 ELSE POKE v_wfd0,PEEK(&HF8D6)
1270 proces$(proces)="CCHANGE.X1":proces=proces+1:CHAIN "GLOAD.X1"
1280 '-----{ SAVE }-----
```

```
1290 '
1300 LABEL "save"
1310 IF proces=0 THEN 1080
1320 LOCATE 7,7:PRINT "*** GRAPHIC SAVE ***"
1330 LOCATE 11,10:COLOR 1:PRINT "[RETURN]: MENU"
1340 LOCATE 5,13:COLOR 6:PRINT "FILE-NAME":COLOR 7:INPUT "",fes(1)
1350 IF fes(1)=" THEN 1080 ELSE POKE v_wfd0,PEEK(&HF8D6)
1360 proces$(proces)="CCHANGE.X1":proces=proces+1:CHAIN "GSAVE.X1"
1370 '-----{ MENU }-----
1380 '
1390 LABEL "menu"
1400 LOCATE 3,5:PRINT "*** 4096 -> 8 color change ***"
1410 LOCATE 10,8:COLOR 3:PRINT "[1]":COLOR 7:PRINT " COLOR CHANGE"
1420 LOCATE 10,10:COLOR 3:PRINT "[2]":COLOR 7:PRINT " GRAPHIC LOAD"
1430 LOCATE 10,12:COLOR 3:PRINT "[3]":COLOR 7:PRINT " GRAPHIC SAVE"
1440 LOCATE 10,14:COLOR 3:PRINT "[4]":COLOR 7:PRINT " END"
1450 LOCATE 14,16:COLOR 5:PRINT "push [1]-[4]"
1460 LOCATE 7,18:COLOR 7:PRINT "[SPACE] : graphic ON & OFF"
1470 k=0:REPEAT:a$=INKEY$:a=VAL(a$)
1480 IF a$=" " THEN IF k THEN SCREEN:k=0 ELSE INIT:k=1
1490 UNTIL (1<=a AND a<=4) OR a$=CHR$(27)
1500 RETURN
1510 '-----{ END }-----
1520 '
1530 LABEL "ending"
1540 CLS:CFLASH 1:PRINT "PUSH SPACE":CFLASH 0
1550 REPEAT:a$=INKEY$:UNTIL a$<>" "
1560 CLS:CONSOLE 0,24:KLIST 1:RETURN
```

リスト2 CCHANGE.OBJ

```
C000 C3 16 F0 00 00 00 00 00 : 99
C008 00 40 01 00 00 00 00 2C : 35
C010 14 2C 00 00 00 00 21 00 : 61
C018 C8 11 01 C8 01 FF 07 AF : 5A
C020 77 ED B0 2A 0B C0 22 05 : 30
C028 C0 D9 DD 21 00 C7 21 00 : 81
C030 C8 D9 3A 05 C0 E6 01 28 : AF
C038 08 D9 E5 D0 E5 E1 DD E1 : 27
C040 D9 D0 E5 E1 5D 54 13 01 : 41
C048 FF 03 AF 77 ED B0 D9 ED : 8B
C050 5B 07 C0 DD 19 19 D9 CD : D7
C058 6B C0 2A 05 C0 23 22 05 : 64
C060 C0 ED 5B 0D C0 B7 ED 52 : CB
C068 38 BF C9 AF 32 0F C0 32 : A2
C070 10 C0 32 11 C0 2A 07 C0 : C4
C078 22 03 C0 C0 8F C0 2A 03 : 2E
SUM: 6E 21 02 C9 15 0A 0E F0 B594
```

```
C080 C0 23 22 03 C0 ED 5B 09 : 19
C088 C9 B7 ED 52 38 ED C9 CD : 71
C090 80 C1 FD 21 0F C0 3E 40 : AC
```

```
C098 32 12 C0 CD 02 C1 FD 7E : 0F
C0A0 00 82 D9 86 D9 16 00 FE : CF
C0A8 78 38 04 D6 78 16 01 CD : E6
C0B0 C7 C0 CD 41 C1 FD 23 DD : 53
C0B8 23 D9 23 D9 3A 12 C0 C6 : CA
C0C0 40 32 12 C0 20 D5 C9 5F : 61
C0C8 CB 3F CB 3F CB 3F FD 77 : 92
C0D0 00 7B E6 07 DD 86 00 FD : C8
C0D8 86 00 DD 77 00 2A 03 C0 : C7
C0E0 7C B5 28 04 FD 7E 00 87 : 65
C0E8 DD 86 FD DD 77 FD 7E : 2C
C0F0 00 DD 86 03 DD 77 03 FD : BA
C0F8 7E 00 CB 27 CB 27 FD 77 : D6
```

SUM: FC 04 AF 47 39 73 09 0E 43E1

```
C100 00 C9 CD 9E C1 16 00 CD : D8
C108 25 C1 01 D0 1F ED 78 F6 : 31
C110 10 FD 79 CD 25 C1 01 D0 : FA
C118 1F ED 78 E6 EF ED 79 7A : 39
C120 07 07 07 57 C9 E5 CD 32 : 19
C128 C1 01 00 04 09 CD 32 C1 : 8F
```

```
C130 E1 C9 4D 44 ED 78 A3 28 : 6B
C138 04 37 CB 12 C9 B7 CB 12 : 75
C140 C9 CD 9E C1 CD 5D C1 01 : E1
C148 D0 1F ED 78 F6 10 ED 79 : C0
C150 CD 5D C1 01 D0 1F ED 78 : 40
C158 E6 EF ED 79 C9 E5 CD 6A : 20
C160 C1 01 00 04 09 CD 6A C1 : C7
C168 E1 C9 4D 44 CB 1A 38 0A : 62
C170 05 78 2F ED 58 A3 ED 79 : CD
C178 D1 C9 FD 78 B3 ED 79 C9 : E1
SUM: 95 B2 80 32 B7 7A CF A3 31E2
```

```
C180 2A 03 C0 ED 5B 05 C0 AF : A9
C188 32 F6 FB 06 1D ED 41 CD : 41
C190 07 59 06 1E ED 41 22 13 : E7
C198 C0 7A 32 15 C0 C9 3A 12 : 56
C1A0 C0 2A 13 C0 84 67 3A 15 : F7
C1A8 C0 5F C9 : E8
SUM: A3 55 CF E6 A9 63 97 B6 5AB4
```


というグラフィック保存用の命令がありますが、GLOAD,GSAVEのデータ形式はそのフォーマットとまったく同じです。したがって、VSAVEによってセーブされたファイルはGLOADでロードできるし、その逆もまたしかりです（違いは「速さ」だけ）。「ベタ書きフォーマット」であり賢くないのですが、これが標準なのでしかたありません。

●GSAVE.X1

グラフィック画面のセーブです。ロードと同じように起動しますが、セーブする前に96Kバイトか64Kバイトにするかを聞いてきます。画像のグラフィックモードに合わせて決定してください。セーブ時もロード時と同じように画面が乱れます。なお、64KファイルはGSAVE独自のものなので、Z-BASICのVLOADではロードできません。

これらのプログラムをKAME-DOS上で使うときには、重要な注意点がひとつあります。よく読んでください。それは、DOSのバッファをG-RAMに設定している場合です（バッファに関しては6, 7月号参照のこと）。画像ファイルをセーブしようとしてディスクアクセスすると、バッファがG-RAM上にあるのでグラフィックが破壊されてしまいます。これでは困るので、バッファをほかに移す必要があるのです。

バンクメモリを搭載していればそこにバッファを設定して一件落着なのですが、そうとばかりは限りません。そこでX1に残された最後の領域であるキャラクタ&アトリビュートエリアに、バッファを設定します（そうです！ このおかげでロード/セーブ時に画面が乱れるのです）。これは一時緊急避難的処置なので、これが終わったらすぐに元へ戻してください。具体的な作業は囲みに書いておきます。

必殺！ アルゴリズム

実は、CCHANGE.OBJ（リスト2）は単独でも使用可能なのです。4096色グラフィックを表示させておいて、CLEAR & HC000:LOADM "CCHANGE.OBJ"でマシン語をロードします。その後、CALL & HC000を実行すれば8色に変換してくれます。これを利用すると、ZでないturboでZのアナログ画像データをロードし（GLOAD）、8色に変換して、アナロググラフィックをそれなりに見ることもできます（データが手に入れば、だけど）。

また、CCHANGE.X1内で、MEM\$(&

HC007,8)=MKI\$(0)+MKI\$(320)+MKI\$(0)+MKI\$(200)という行があります。このMKI\$の中身は順に左上X座標、右下X座標+1、左上Y座標、右下Y座標+1になっていて、この矩形領域が変換対象になります。書き換えて実行してみるとよくわかると思います。

さて、CCHANGEルーチンのアルゴリズムはバックナンバーを見てもらうとして、ここではGLOADとGSAVEについて解説します。

それぞれKAME-DOSのディスクアクセスルーチンを使っているわけですが、KAME-DOSには標準のG-RAMロード&セーブルーチンはありません。どうしてるのかというと、G-RAM全体(48Kバイト×2)をバッファとみなして、通常はデータの仲介役のバッファに、最初からデータを入れておいたことにします。

もともとバッファの大きさは4Kバイト単位の変長なので48Kバイトでも問題はありません。

この方式の長所としては「BASICでも簡単に制御できる・速くなる」などがあって、短所は「ベタ書きフォーマットにしか通用しない・任意の矩形領域は取り扱いできない」などです。そのためZ'STAFFのフォー

図1 拡張子と画面モードの関係

[X:] GLOAD A:GAZO.GLO

```

GLO: WIDTH 40,25,0,1 4096色モード
GL1: WIDTH 80,25,0,1 64色モード
GMO: WIDTH 40,25,0,2 64色モード
GMI: WIDTH 80,25,0,2 8色モード
GHO: WIDTH 40,25,1,2 64色モード
GHI: WIDTH 80,25,1,2 8色モード
GL2: WIDTH 40,25,0,1 64色2画面

```

(マニュアルより抜粋)

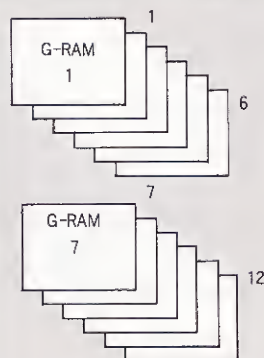
これに新たに、GL3: WIDTH 40,25,0,1 8色モードをつけました。なおGL3の場合は32Kバイトしか使っていませんが、64K分のファイルになります。その他はすべて96Kファイルになります。

マットとは互換性がありません。

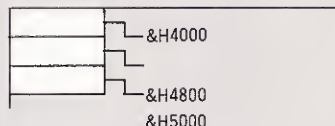
* * *

今回は、特集がメインなのか連載がメインなのかかわからなくなってしまいました。ただ、Zに関してはマウス・画像取り込みなど手つかずになってしまったのが残念です。CCHANGEなどはちょっとした変更でまだまだ拡張できるプログラムなので、やりたいことを自分でプログラミングしてみましょう。

図2 X1のG-RAM構成



X1 turbo (320×200) のG-RAM構成は上記のような12枚構造です（ノーマルX1は6枚のみ）。それぞれのG-RAMの左上のアドレスは、&H4000,&H4400,&H8000,&H8400,&HC000で、7から12は裏バンクになっています。



G-RAMの左上を拡大したもので、縦に8段あって、それぞれ&H800ごとのアドレスに割りふられています。

X1turboZでも同じような構成になっていますが、4096色の場合はG-RAM12枚をまとめて2¹²=4096を表しています。青はG-RAMの1,2,7,8・赤は3,4,9,10・緑は5,6,11,12です。BRGそれぞれについて4ビット16階調、かたや8色は1ビット1階調（あるか、ないかだけ）です。

バッファの設定の仕方

グラフィック用バッファを確保する場合もINTEGRAL.Xを書き換えます。

1) Z-BASICの場合

```

I210 MEM$(S_FF,2)=MKI$(&H5000):
MEM$(S_BUFF,2)=MKI$(&H6000)
I220 MEM$(S_BSIZ,2)=MKI$(&H1000):
POKE S-IOMM,4
に差し換えてください。

```

2) バンクメモリがない機種すべて

```

I210 MEM$(S_FF,2)=MKI$(&H2000):
MEM$(S-BUFF,2)=MKI$(&H3000)
I220 MEM$(S_BSIZ,2)=MKI$(&H1000):

```

POKE S-IOMM,1

KAME-DOS上からCCHANGE,GLOAD,GSAVEを使う場合は上記のようにする必要があります。さらに、ノーマルX1の場合は次の1行も付け加えてください。

```
I225 POKE &HE139,8
```

2)の書き換えを行ったINTEGRAL.Xでは、CCHANGE,GLOAD,GSAVEの立ち上げ以外は行わないようにしてください。「DIR」などをすると、画面がメチャクチャになってしまいます。もとへ戻すには、書き換えを行う前のINTEGRAL.Xを起動しなおしてください。

リスト3

```

0000      1
0000      2
0000      3
0000      4
0000      5
0000      6      ORG $C000
0000      7
FBFE P    8      #SCRNM2 EQU $FBFE
5907 P    9      #GRAADR EQU $5907
C000 P   10      #LCP EQU $C000
C000 P   11      #LCN EQU $C000
C000      12
C000 C3 16 C0 13      JP BEGIN
C003      14
C003      15
C003 00 00 16      X      D# 0
C005 00 00 17      Y      D# 0
C007 00 00 18      X1     D# 0
C009 40 01 19      X2     D# 320
C00B 00 00 20      Y1     D# 0
C00D C8 00 21      Y2     D# 200
C00F      22      BRGB DS 3
C012 00      23      RGBP DS 0
C013 00 00 24      ADR     DW 0
C015 00      25      BIT     DB 0
C016      26
C016      27
C016      28      BEGIN
C016 21 00 C8 29      LD HL,#LCP
C019 11 01 C8 30      LD DE,#LCP+1
C01C 01 FF 07 31      LD BC,$7FF
C01F AF      32      XOR A
C020 77      33      LD (HL),A
C021 ED 00      34      LDIR
C023 2A 0B C0 35      LD HL,(Y1)
C026 22 05 C0 36      LD (Y1),HL
C029      37      FORN
C029 D8      38      EXX
C02A DD 21 00 C0 39      LD IX,#LCN
C02F 21 00 C0 40      LD HL,#LCP
C031 D9      41      EXX
C032 3A 05 C0 42      LD A,(Y)
C035 E6 01      43      AND I
C037 28 08      44      JR Z,FINVHL
C039 D9      45      EXX
C03A E5      46      PUSH IX
C03B DD E5      47      PUSH IX
C03D E1      48      POP HL
C03F DD E1      49      POP IX
C040 D9      50      EXX
C041      51      FINVHL
C041 DD E5      52      PUSH IX ;LC(C,N,X)
C043 F1      53      POP HL
C044 5D      54      LD F,L
C045 54      55      LD D,H
C046 13      56      INC DE
C047 01 FF 03 57      LD BC,$7FF
C04A AF      58      XOR A
C04B 77      59      LD (HL),A
C04C FD 00      60      LDIR
C04F D9      61      EXX
C04F ED 5B 07 C0 62      LD DE,(X1)
C053 DD 19      63      ADD IX,DE
C055 19      64      ADD HL,DE
C056 D9      65      EXX
C057 DD 6A C0 66      CALL VLP
C05A 2A 05 C0 67      LD HL,(Y)
C05D 23      68      INC HL
C05F 22 05 C0 69      LD (Y),HL
C061 FD 5B 0B C0 70      LD DI,(Y2)
C065 07      71      OR A
C066 FD 52      72      SRC HL,DE
C06A 3F 0F      73      JR C,FORX
C06A C9      74      RET
C06B      75
C06B      76      XLP
C06B AF      77      NOP A
C06F 32 0F C0 78      LD (BRGB),A
C06F 32 10 C0 79      LD (BRGB+1),A
C072 32 11 C0 80      LD (BRGB+2),A
C075 2A 07 C0 81      LD HL,(X1)
C078 22 03 C0 82      LD (X1),HL
C07B      83      FORN
C07B CD 0F C0 84      CALL R03
C07E 2A 03 C0 85      LD HL,(X)
C081 23      86      INC HL
C082 22 03 C0 87      LD (X),HL
C085 ED 5B 05 C0 88      LD DE,(X2)
C089 B7      89      OR A
C08A ED 52      90      SRC HL,DE
C08C 38 ED      91      JR C,FORX
C08E C9      92      RET
C08F      93
C08F      94      R03
C08F CD 80 C1 95      CALL XYADR
C092 FD 21 0F C0 96      LD IX,BRGB
C096 3E 40      97      LD A,$40
C098 32 12 C0 98      LD (ROBP),A
C09B      99      RGBLP
C09D CD 02 C1 100      CALL POINT
C09F FD 7E 00 101      LD A,(IX+0)
C0A1 82      102      ADD A,D
C0A2 D9      103      EXX
C0A3 86      104      ADD A,(HL) ;LC(R,P,X)
C0A4 D9      105      EXX
C0A5 16 00      106      LD D,0
C0A7 FE 78      107      CP 120
C0A9 38 04      108      JR C,SIXII
C0AB D6 78      109      SUB 120
C0AD 15 01      110      LD D,I
C0AF      111      SIXII
C0AF CD C7 C0 112      CALL LCINC
C0B2 CD 41 C1 113      CALL PSET
C0B5 FD 23      114      INC IX
C0B7 DD 23      115      INC IX
C0B9 D9      116      INC IX
C0BA 23      117      INC HL
C0BB D9      118      EXX
C0BC 3A 12 C0 119      LD A,(RGBP)
C0BF C6 40      120      ADD A,$40
C0C1 32 12 C0 121      LD (ROBP),A
C0C4 20 D5      122      JR NZ,RGBLP
C0C6 C9      123      RET
C0C7      124
C0C7      125      LCINC
C0C7 5F      126      LD E,A
C0C8 CB 3F      127      SRL A
C0CA CB 3F      128      SRL A ;A/8
C0CC CB 3F      129      SRL A
C0CE FD 77 00 130      LD (IX+0),A
C0D1 78      131      LD A,E ;DOWN
C0D2 E6 07      132      AND 7
C0D4 DD 06 00 133      ADD A,(IX+0) ;LC(R,N,X)
C0D7 FD 06 00 134      ADD A,(IX+0)
C0DA DD 77 00 135      LD (IX+0),A
C0DD      136
C0DD 2A 03 C0 137      LD HL,(X)

```

```

C0E0 7C      138      LD A,E
C0E1 35      139      OR L
C0E2 28 0A      140      OR C,BRBLD
C0E4 FD 7E 00 141      LD A,(Y) ;EFG+DOWN
C0E7 57      142      OR A,A
C0E9 DD 46 00 143      LD A,(Y+1) ;LC(R,N,X+1)
C0EB DD 77 00 144      LD (Y+2),A
C0EF      145      RET
C0EE      146
C0EE FD 7E 00 147      LD A,(Y) ;RIGHT+VHL
C0F1 DD 5B 01 148      LD BC,$7FF
C0F4 DD 77 00 149      LD (Y+2),A ;LC(R,N,X+1)
C0F7      150
C0F7 FD 7E 00 151      LD A,(Y) ;RIGHT
C0FA CB 21 152      OR A,A
C0FC CB 21 153      OR A,A
C0FE DD 77 00 154      LD (Y+2),A
C101 C9      155      RET
C102      156
C102      157      POINT OUT 2
C102      158
C102 CD 0E C1 159      CALL RETBLE
C105 16 00      160      LD D,0
C107 CD 25 C1 161      CALL BTM00
C10A 01 0B 1F 162      LD BC,$7FF
C10D ED 75      163      INC A
C10F F5 10      164      OR S08
C111 ED 79      165      ORT A,A
C113 CD 25 C1 166      CALL BTM00
C116 01 0B 1F 167      LD BC,$7FF
C119 ED 75      168      INC A
C11B E6 07      169      AND 6EF
C11D ED 79      170      ORT A,A
C11F 7A      171      LD A,D
C120 07      172      S00A
C121 07      173      S00A
C122 07      174      S00A
C123 57      175      LD D,A
C124 C9      176      RET
C125      177
C125      178      BTM00
C125      179
C125 E5      180      PUSH HL
C126 CD 31 C1 181      CALL BTM00
C129 01 00 04 182      LD BC,$400
C12C 89      183      ADD HL,BC
C12D CD 32 C1 184      CALL BTM00
C130 E1      185      POP HL
C131 C9      186      RET
C132      187
C132      188      BTM00
C132 CD 21 C1 189      LD D,C
C133 44      190      LD E,E
C134 ED 79      191      INC A
C136 A3      192      AND E
C137 28 04      193      OR D,ROBIT
C139 37      194      S00F
C13A CB 12      195      OR C
C13C C9      196      RET
C13D      197
C13D      198      ROBIT
C13D B7      199      OR A
C13E CB 12      200      OR C
C140 C9      201      RET
C141      202
C141      203
C141      204      RET ON 2
C141      205
C141 CD 0E C1 206      CALL RETBLE
C144 CD 5D C1 207      CALL BTM00
C147 01 0B 1F 208      LD BC,$7FF
C14A ED 78      209      INC A
C14E F5 10      210      OR S08
C14F ED 75      211      INC A
C150 CD 5D C1 212      CALL BTM00
C153 01 10 1F 213      LD BC,$7FF
C156 ED 78      214      INC A
C158 E6 07      215      AND 6EF
C15A ED 7A      216      ORT A,A
C15C C4      217      RET
C15D      218
C15D      219      BTM00
C15D E5      220      PUSH HL
C15E CD 6A C1 221      LD BC,$400
C161 01 00 04 222      LD BC,$400
C164 09      223      AND HL,BC
C165 CD 6A C1 224      CALL BTM00
C168 F1      225      S00F
C169 C9      226      RET
C16A      227
C16A      228      BTM00
C16A 4D      229      LD C,D
C16B 44      230      LD E,E
C16C CB 1A      231      OR C,B
C16E 36 0A      232      OR C,BT0A
C170 C4      233      BTM00
C171 78      234      LD A,E
C172 2F      235      CPL
C173 ED 78      236      INC A
C175 A3      237      AND E
C176 DD 75      238      ORT A,A
C178 D1      239      POP DE
C179 C9      240      RET
C17A      241      BTM00
C17A ED 78      242      INC A
C17C B3      243      OR E
C17D ED 79      244      ORT A,A
C17F C9      245      RET
C180      246
C180      247
C180      248      XYADR :IN X,Y OUT ADR,BIT
C180 2A 03 C0 249      LD HL,(X)
C183 ED 5B 05 C0 250      LD BC,(Y)
C187 A3      251      OR A,A
C188 32 F6 FB 252      LD (#SCRNM2),A
C18B 06 10      253      LD S,$10
C18D ED 43      254      ORT A,A
C18F CD 87 C9 255      CALL #GRAADR
C192 06 1E      256      LD S,$1E
C194 ED 40      257      ORT A,A
C196 32 13 C0 258      LD ADR,HL
C199 7A      259      LD A,D
C19A 32 15 C0 260      LD BIT,A
C19D C9      261      RET
C19E      262
C19E      263
C19E      264      SETBLE :OUT HL,E
C19E 3A 12 C0 265      LD A,(ROBP)
C1A1 2A 13 C0 266      LD HL,(ADR)
C1A4 84      267      ADD A,H
C1A5 67      268      LD R,A
C1A6 3A 15 C0 269      LD A,(BIT)
C1A9 5F      270      LD F,A
C1AA C9      271      RET
C1AB      272

```


X68000用画像回転プログラム

XROT0.X

Watanabe Shinya

渡辺 伸也

皆さんこんばんは。拡大縮小回転というと、現在のビデオゲームを語るうえでのひとつのキーワードになっていますね。僕もアフターバーナーに感動してからパソコンでもこういうことができないかなーと思い始めました。

アフターバーナー以前にもA-JAXというものがあつたようですが、僕がゲーセンに顔を出すようになったのはアフターバーナーの出る少し前あたりからなのでA-JAXの存在すらX68000への移植の話が持ち上がるまで知りませんでした（ちなみにそれ以前に最後にゲーセンに行ったときはたしかクレイジークライマーとかがあつて、任天堂のゲームウォッチが流行り始めていた頃だったような）。

アフターバーナーの出た頃といえば2年以上前の話。そんな長いことかかってこのプログラムを作っていたわけではもちろんありません。このプログラムの原形は1年ほど前にすでにありましたが、「スピードを追求するあまり、画像がやたら汚い」、「エラーチェックをしないので、ちょっと使い方を間違っただけですぐにバスエラーが出る（今回投稿したこのプログラムではさらにパワーアップしていてハングアップする危険すらある）」などいろいろなアラがあつて投稿作品としては失格だと考え、投稿は断念したのでした。

その少しあとにGROT.Xを目にして、「この分だと近いうちに誰かがプログラムを発表してX68000ユーザーにとって回転アルゴリズムは一般的なものになるであろう」と読んでいました。が、そうなる気配はない。アフターバーナーのX68000版が出たとき、かねがね気になっていた回転プログラムを調べてみますと（電波さんにケチをつけるつもりはありませんが）僕が開発していた過程のアルゴリズムではありませ

んか。そんなわけで「こういう性格の作品が受け入れられるかどうか一度Oh!Xの読者と編集部挑戦してみるか」と考えまして、

今回の投稿へと至ったのです。

注意事項

何度もいいますがこのプログラムはスピードだけがウリで、絵はボロボロ。これで各種エラーチェック機能をつけて遅くならうものならどーにもこーにも救いがないので、エラーおよび不都合な動作に関する責任はどうしてもユーザーに負ってもらうことになります。

けど、ユーザー側に責任を負わせるソフトは悪いソフトだなんていえませんよね。グラフィックツールや各種言語、特にアセンブラ。よっぽどタコなソフトでなければ作品の不出来をソフトのせいにはしませんね。

マシン語なんか暴走するのが常（？）だから、誰もがテストランする前にRAMディスクの内容をセーブするし、大事なファイルの入ったディスクを実験に使ったりはしません。これはアセンブラに関しては「不都合が起った場合の責任は自分にある」という認識が一般に広まっているからなのです。暴走したとき、真っ先に考えるのは「自分のミス」であつて、「こんなはずはない！アセンブラがバグっているのだ」なんてチラつとも考えないはずです。

僕としてはその辺が不安の材料なわけで、「暴走するグラフィック関数」なんて皆さんはいままで見たことも聞いたこともないと思います。でもこのプログラムがそうなんです。くれぐれも注意してください。これのせいで大事なファイルが消えてしまったなんてことがないように。取り越し苦労でしょうか。

遊び方

リストはできるだけ多くの方が打ち込む気になってくれるように短くしたつもりです。まず、リストを入力します。コンパイルは、

読者投稿による画面回転プログラムです。比較的小さなリストでも効果てきめん。256色の画像をグルグル回します。特殊効果その他、画像処理の際に参考にしてください。なお、高速化のためエラー処理など一部省略された処理がありますので注意してください。



CC /W XMKDAT0.C

です。そして実行ですが、このプログラムはカレントディレクトリに約30Kバイトのファイルを作成しますので、カレントにはその分の余裕が必要です。

プログラムを実行すると放射線が描かれ、中心が抜けていきます。放射線は360度制で3度おきに120本。抜けた中心部分は直線データとして、ファイルに吸収したのです。そしてそのファイルが出力されるXROT DAT0です。

あとはXROT0.SとTESTROT.Cを入力して、

CC /W /Y /w TESTROT.C
XROT0.S

とすればTESTROT.Xができます。

実行時にはXROT DAT0がカレントにあるようにしてください。また、実行する前に16色または256色モードの絵をページ0にロードしておきます。絵は512×512ドットいっぱい描き込んであるものを選んでください（65536色のデータはto256.xなどで変換してください）。

適当なデータがないときはTESTROT.Cのコメント化してある行を有効にしましょう。操作方法は図1ですが、まずはこちらの指示に従って操作してください。では実行です。

TESTROT

まずOPT.1キーで、縮小していきます。するとすぐに画面はページ0の大きさを越え、本来の画像の周囲に変な画像が出現します。そのあたりで操作を止めて、ページ

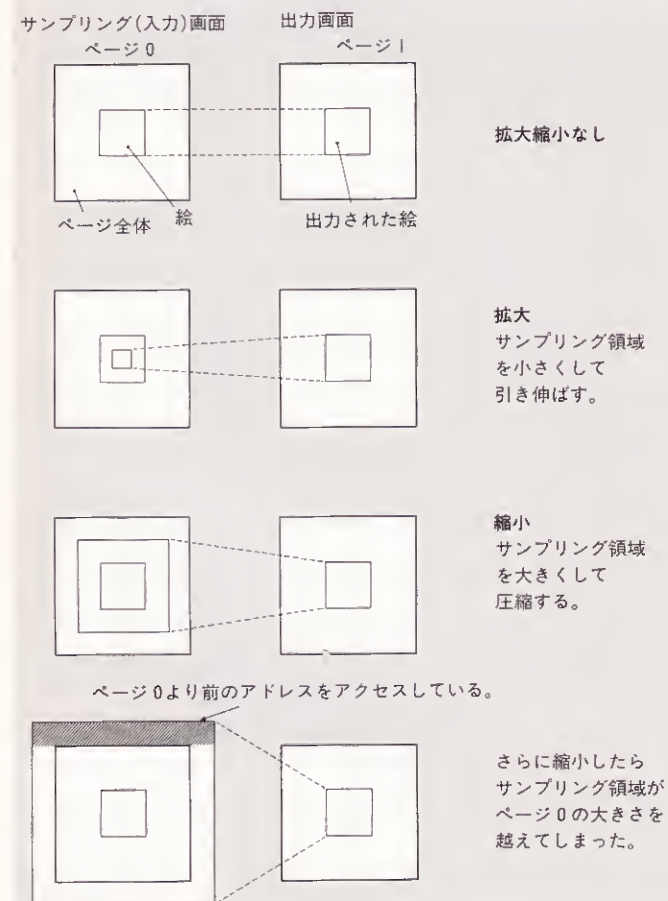
0の大きさを越えた画面上部の黒い部分に注目してください。拡大縮小のアルゴリズムは図2です。

この黒い部分はページ0のアドレス(\$C00000)より前の領域からデータを読んでいることがわかります。しかしRAMをフル装備したマシンでない限りこの領域にメモリは存在しないわけで、普通この領域をアクセスすれば「バスエラーが発生しました」となるのですが、いまそうならないのは画像を作成しているあいだだけ、バスエラーベクタを書き換えてオリジナルのバスエラー例外処理プログラムで処理しているからなのです。このへんは実際にリストを打ち込んだ方なら察しがついていると思います。

ほとんどのユーザーのマシンでこの黒い部分の面積に比例して処理が遅くなりますが、これはバスエラーの数だけ例外処理にとんでいるためです。

絵が左右に連なっているのも含めて、このプログラムでは「絵からはみ出したかどうかチェックして回避しない」のが諸悪の根源なわけですが、その処理を入れると極

図2 拡大縮小



端に遅くなってしまうのです。

ところで画面の下の方に見えている「変な画像」についてですが、まずTEST TROT.Xの動作を理解してください。図3です。

縮小するとサンプリング領域が広まってページ0の上をサンプリングしてバスエラーを出しますが、ページ0の下もサンプリングします。ページ0の下とはページ1、

図1 TEST TROT.Xのキーボード操作

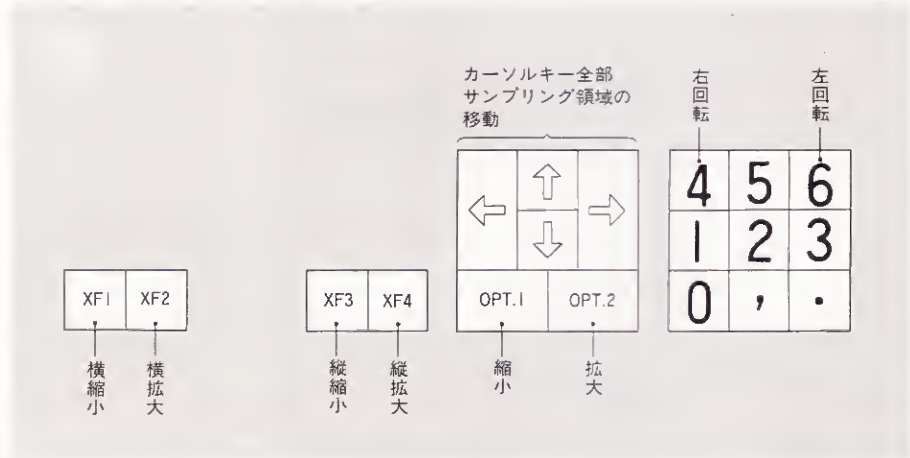
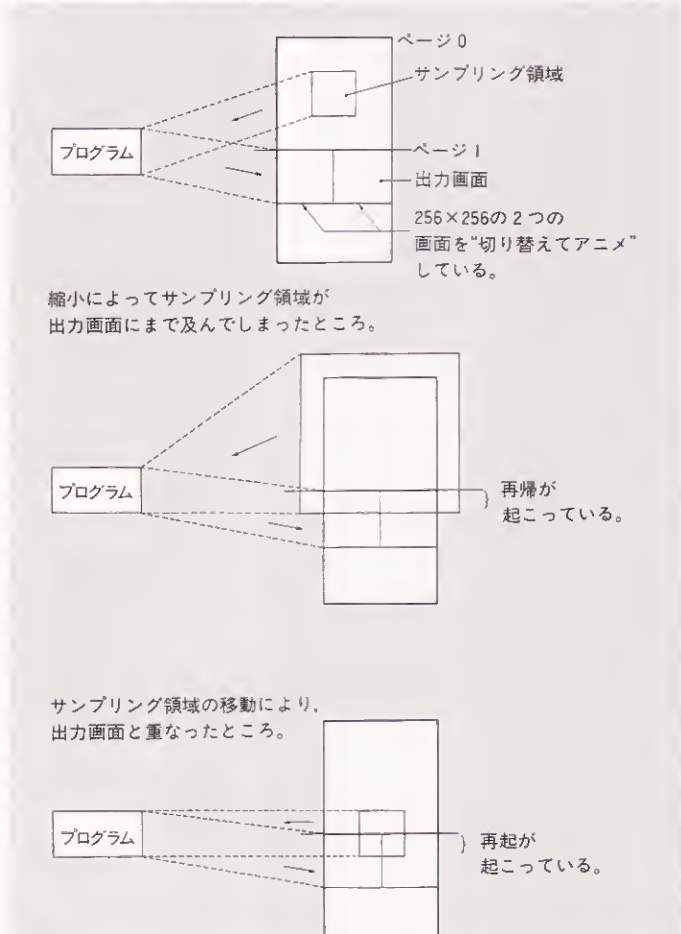


図3 TEST TROT.Xの動作



ひと通りいじくって使い勝手をのみこんだらXROT0.Sのバスエラー回避部分を切ってしまうでしょう。さすがにこれは危なすぎるので。描画処理中に割り込んできたプログラムがバスエラーを起こした場合にも気まづいものがあるし。あと、このバスエラー回避は読み込み側画像のエラー処理専門なので、それ以外のバスエラーだとハングアップします。

以後、その絵やページからはみ出すようなパラメータの設定をしないように気をつけましょう。

関数の説明

XROT0というまでもなくXC対応に作られた関数で、数Kバイトのプログラム部と、約30Kバイトのデータ部に分けられます。実際にもデータ部はXROTDAT0として別個に存在していて、プログラムが立ち上がったあとに読み込みます。本来プログラムとデータは一体化していました。これはほかのOS上（自作とか）でも動かすことを想定していたからで、Humanのコマンド（この場合はディスク関係）は使いたくなかったのです。しかしそれだとリンクが長いので今回の投稿では作り直しています。

なおXROTDAT0のあるディレクトリはXROT0.Sのデータ文で決定されるので、ここを書き換えてしまえば、どこにファイルがあろうとかまいません。

XROT0は以下の3つの関数から成り立っています。

XROT0INIT();

ディスクからデータを読み込みます。プログラムが立ち上がったら、1回実行してください。実行しなかったら素直にバグるのみです。純正のグラフィックコマンドは「画面初期化コマンドを実行していない」ことを察知するとシカトしてくれますね。IOCSレベルからしてそういう構造になっていますが皆さんはこういうのを親切な設計だと思いますか？

IOCSといえばあのレジスタをビシバシ使うやり方はいただけません。一度作った値をメモリに待避しないでそのままレジスタに残しておいて使えるのが68000のプログラミングスタイルであり、68000の価値だと考えます。Cのようにパラメータをスタックで渡すとか、パラメータ群のセットしてあるメモリの先頭ポインタをスタックで渡すとかのほうがよいと思うのですがどうでしょう。アセンブラで組むときはまずスーパーバイザモードにして、I/Oポートの

操作は自力でやる人って多いと思います。

WNDROT0(P0,P1);

INT P0:

入力画像のあるページ番号(0~3)

INT P1:

画像を出力するページ番号(0~3)

P0=P1であってもかまわない。負数禁止。

実際にプログラムがほしがっているのは各ページ番号ではなくて、各ページの座標(0,0)のアドレスです。それをP0,P1より計算して内部に控えておきます。計算式は、

アドレス = \$C00000 + \$80000 × ページ番号

です。0~3以外の値も受け付けます。たとえば4だと\$E00000、つまりテキストVRAMを指定できます。XROT0は拡大縮小回転しなければ、ただの画像転送命令として使えるので、たとえば、

WNDROT0(0,4);

としてテキストVRAMに絵を置いておき、以後、

WNDROT0(4,0);

とすれば、グラフィックVRAMに入力画像を置いておく必要はなくなります。

ただし、XROT0はメモリのどこを指定しても実画面512×512ドットのフォーマット(X方向のカウン트가±2バイト、Y方向のカウン트가±1024バイト)として扱うので、16色モードの絵を1ページ分転送してもテキストVRAMの全部を使ってしまいます。

XROT0(X1,Y1,X2,Y2,W,H,SX,SY,A);

int X1: 入力画像の中心のX座標

int Y1: 入力画像の中心のY座標

int X2: 出力画像の中心のX座標

int Y2: 出力画像の中心のY座標

int W: 出力画像の横サイズの1/2

int H: 出力画像の縦サイズの1/2

int SX: 横の拡大縮小率(下位2バイトのみ有効)

int SY: 縦の拡大縮小率(下位2バイトのみ有効)

int A: 回転角度 ±3900000

中心の座標とはいわずと知れたその画像の縦と横の中心に位置するドットの座標ですが、中心の1ドットが存在するためには画像のサイズが縦横ともに奇数でなければなりません。

しかるに画像のサイズは1/2の状態で指定するので、サイズは必ず偶数になり真の中心ドットはなくて代わりに、候補の4ドットが存在するかたちになります。ではどうするのかというと、4ドットのうちの任

意の1ドットを中心として決めてしまっただけです。どうせ精度はガタガタなので1~3ドットの違いなど問題になりません。

ですが拡大縮小なしで回転角度が0, 30, 60, 90, 120の倍数のときはさすがに正確に動きますのでそのあたりも考慮してください。

サイズを1/2で指定させるのはこの値で計算することが多いのと、1/1の値から1/2の値を作ろうとすると、奇数だった場合に誤差が出てしまうからです(整数演算なので)。精度はガタガタだと書きましたが、なるべくそうならないようにはしているのです。

拡大縮小率の設定はややこしくて、任意の拡大縮小率の逆数を16進数の固定小数点小数として考えます。小数点は下位2バイトのあいだにとって、\$00.00とします。

概念としては、

2倍 拡大 → 1/2 → \$01.00/2 → \$00.80 → \$0080

1/2 縮小 → 2/1 → \$01.00 * 2 → \$02.00 → \$0200

となります。

実用的な計算法は、

double A = 1.25;

/* 実数によるわかりやすい表記 */

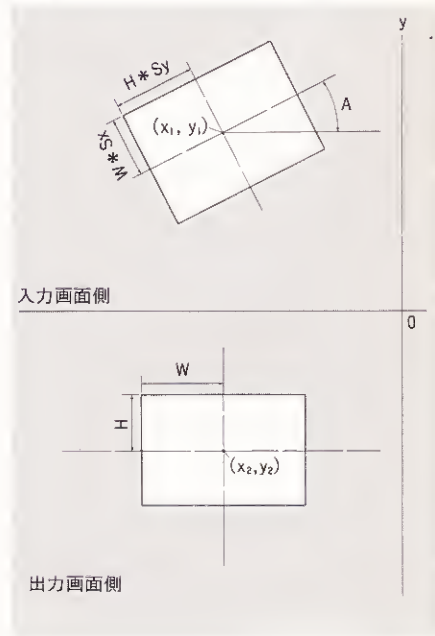
int SX,SY;

SX = (int)(1.0 / A * 256.0);

です。

なお、いまの状態では縮小率は1/4までですが、この制限を加えているのは、XROT0.Sの18行と19行だけですので変更するのは簡単です。回転角度の制限はDIVSによる

図4



割り算の限界の事です。

回転のアルゴリズム

いろいろな回転プログラムが出まわっていますが、どれも似たような作りをしています。僕は非常に奥が深いものかと思ってさまざまな試みをしました。その結果、誰もが最初に思いつくアルゴリズムが実は一番優秀であることがわかってきました。なんか残念です。

簡単に説明すると、

「目的の画像より水平に1ライン分のデータをサンプリングし、それに回転処理を施し目的のVRAMに出力する。これを縦の大きさだけ繰り返す」

「目的の画像より回転処理を施したライン上（つまり水平垂直を含むナナメ線）に1ライン分サンプリングし、それを水平に変換して目的のVRAMに出力する。これを縦の大きさだけ繰り返す」

わけです。前者を画像回転型、後者を座標回転型と僕は呼んでいます。

この投稿は後者の座標回転型であり、描画面積が一定なため「描画速度が一定で次の絵を出すとき、前の絵をクリアすることなくそのまま上書きしてしまえる」というアニメ処理に好都合な特徴を持っています。

画像回転型は描画面積が不定なので、同じ画像データでも縮小しているほど速く処理が終わるような作り方が可能です。あるいは工夫が足りないと、拡大縮小によって処理速度が違ってくるような作りになってしまいがちです。またアニメ処理のときはいちいち前の絵をクリアする必要があります。こう書くと座標回転型に比べて悪いことづくめようですが、小規模な処理に向いているのでゲームにはこちらのほうがよく採用されるようです。

2つのアルゴリズムは実はただ単にデータの流れが逆になっているだけだ、ということに気がつくられましたか？ この2つは基本的には同じひとつのアルゴリズムです。

いわれてみれば簡単な構造でアルゴリズムのうちに入らないかもしれません。回転処理の難しさはプログラムそのものよりも処理の概念やイメージをつかみにくいところにあるように思います。

スピードアップのポイントは1ラインの転送をいかに速くするかにかかっており、転送は普通ループで処理しているのでこれを展開します。展開すると画面の横幅が固定されますが、このプログラムでは自己書き換えによって1ライン分の転送ルーチン

を「作る」ので可変長になっています。

また、この転送ルーチンの中で、

LEA d16(An),An

が使われています。これは最高速の32ビット加減算命令なのですが、イミディエイトであるd16の部分はメインメモリ上の値なのでこれを変更したくば、または自己書き換えです。よってこのプログラムではひとつの領域に2カ所から書き換え動作を行うことによって、ひとつの転送ルーチンを作り出しています。

XROT0は1ドット/20クロックの描画速度を持っているので、256×256ドットの画像だと秒速7コマで書き換えることがで

きます。いろいろとサイズを変えて実験してみましょう。

あと注意が必要なのはVPAGE, HOMEなどの関数で、これらは垂直帰線期間を無視して動くので、垂直帰線期間待ちをする処理が別個に必要だということです。TES TROT.Cではasmでやっている部分です。Cで作ることもできますが、アセンブラで2行だと知っているCを使う気になりません。

XROT0の拡大回転処理はウソ臭いですがね。本当だったらOh!FM3月号の「view.exp」のように拡大した四角い1ドットにも回転処理を加えなければならないところ

回転について

までも通常座標の方便なので、そこに大きな無理が生ずる。それはドット画面に真の斜線が描けないのと同じである（階段になってしまう）。したがってドット構成の画面である限り、真の画像回転は不可能であり、すべて疑似的なものになる。ハードウェアによる回転でもその例にもれずチラついている。

回転処理に使う画像データはなるべくチラつきを目立たなくするためにグラデーションを多用してボカシ気味に描くのがコツである。ゲーセンに行って確かめてみよう。本当に綺麗に回転させたくば何千色も使って色の補間をする処理が必要である。

リスト1

```
===== XMKDAT0.C =====
1: #include "basic0.h"
2: #include "BASIC.h"
3: #include "graph.h"
4: #include "math.h"
5: #include "stdio.h"
6:
7: main()
8: {
9:     int a,b,c,sampx,sampy,ax,ay;
10:    short int x[128];
11:    double DEG,SC,CO;
12:    FILE *fi;
13:    DEG = pi() / 180.0;
14:    screen( 2, 0, 1, 1 );
15:    home( 0, 128, 128 );
16:    window( 0, 0, 1023, 1023 );
17:
18:    fi = fopen( "XROTAT0", "wb" );
19:    for( a = 0; a < 360; a += 3 ){
20:        ax = (int){ 370.0 * cos( (double)a * DEG ) };
21:        ay = (int){ 370.0 * sin( (double)a * DEG ) };
22:        line( 512, 512, 512 + ax, 512 - ay, 10, 'NASI' );
23:        sampx = 512; sampy = 512;
24:        c = a / 90; c = a - c * 90;
25:        if( c > 45 ) c = 90 - c;
26:        CO = 1.;
27:        SC = cos( (double)c * DEG );
28:        for( b = 0; b <= 127; b++ ){
29:            if( point(sampx, sampy - 1) == 10 ){ ax = 0; ay = -1; }
30:            if( point(sampx + 1, sampy - 1) == 10 ){ ax = 1; ay = -1; }
31:            if( point(sampx + 1, sampy ) == 10 ){ ax = 1; ay = 0; }
32:            if( point(sampx + 1, sampy + 1) == 10 ){ ax = 1; ay = 1; }
33:            if( point(sampx, sampy + 1) == 10 ){ ax = 0; ay = 1; }
34:            if( point(sampx - 1, sampy + 1) == 10 ){ ax = -1; ay = 1; }
35:            if( point(sampx - 1, sampy ) == 10 ){ ax = -1; ay = 0; }
36:            if( point(sampx - 1, sampy - 1) == 10 ){ ax = -1; ay = -1; }
37:
38:            CO = CO * SC;
39:
40:            if(CO > 1.) {
41:                CO = CO - 1.;
42:                pset(sampx,sampy,0);
43:                sampx += ax; sampy += ay;
```


をXROT0では単にソフト的にドットを粗くしただけだったりします。

ところで、このウソ臭い回転、あのアフターバーナー（もちろん本物）がやっているのを知っていますか？ ここからは僕の憶測ですが、アフターバーナーのハードはそれまでのセガの体感シリーズであるスぺハリ、エンデューローサー（マイナー）と同じで、スプライトには拡大縮小機能しありませんでした。

アフターバーナーは2MバイトのRAM（と聞いた）を増設し、そこに回転パターンをこさえてパターン持ちの回転処理をするというパソコンライクな作りをしていたのです。

改造のポイント

XROT0では画像を小さく設定すれば当然処理が高速になります。が、もうひとつ高速化する方法があります。それはソースリストを書き換えることになりませんが、XROT0.Sの、

197行を有効にする

200行を有効にする

201行を無効にする（コメント化する）

210行を有効にする

ことです。どうです？ なかなかうまいことやったと思いませんか？

しかし本当はプログラムを皆さんが理解して、そのうえで自力で改造していただくのが理想です。ですが本気でプログラムを解説すると何ページあっても足りないのそれは諦めました。

その他諸々

回転というすぐに「アサルト！」とか「ダートフォックス（メタルホークでない）ところがポイント。このゲーム好きなんですけど廃れるのが早いのです。しくしく。CD買いました）の移植だ！」とか聞こえてきそうですが、それは無理というものです。

処理速度の問題もありますがここを強調したいのです。XROT0は「1枚絵の回転」ですが、ナムコのシステム2は「BG画面の回転」です。BGとはX68000に搭載のあのスプライトBGのことです。ですからまるっきり違うのだということを理解してください。多くの人は回転ばかりに気を取られているようですけど。

しかし自分でプログラムを組もうとでもしてみない限りそこまで考えないのは当然

```
44:         }else{
45:             ax = ay = 0;
46:         }
47:         x[b] = 2 * ax + 1024 * ay;
48:     }
49:
50:     fwrite( (char *)&x, 2, 128, fi );
51:     printf( "%d\n", x[127] );
52: }
53:
54: for( a = 0; a <= 359; a += 3 ){
55:     putw( ( short int )(sin( a * DEG ) * 4096.0 ), fi );
56:     putw( ( short int )(cos( a * DEG ) * 4096.0 ), fi );
57: }
58: fclose( fi );
59: screen( 2, 0, 1, 1 );
60: }
```

リスト2

```
===== TESTROT.C =====
1: #include "basic0.h"
2: #include "graph.h"
3: #include "doslib.h"
4: #include "iocslib.h"
5:
6: main()
7: {
8:     int a,b,c,d,SSP;
9:     int X1,Y1,X2,Y2,W,H,SX,SY,A;
10:    int a_frag;
11:
12:    SSP = B_SUPER(0);
13:    C_CUROFF(); A_CLR_AL();
14:    CRTMOD( 10 + 0x100 );
15:
16:    /*****S A M P L E*****/
17:    screen( 0, 2, 1, 1 );
18:    window(0,0,511,511);
19:    apage(0);
20:    for(a = 0; a < 16; a++ ) palet( a, rgb( 2*a, 2*a, 2*a ) );
21:
22:    fill( 0, 0, 511, 511, 15 );
23:    fill( 10, 10, 501, 501, 0 );
24:
25:    for(a = 0; a < 511; a += 64){
26:        for(b = 0; b < 511; b += 64){
27:            for(d = 0; d < 23; d +=4){
28:                c = ( a + b + c ) - ( ( a + b + c ) / 15 ) * 15;
29:                fill( a+d, b+d, a+50-d, b+50-d, c );
30:            }
31:        }
32:    }
33:    box( 256-128, 256-128, 256+128, 256+128, 15, 'NASI' );
34:    fill( 256-40, 256-40, 256+40, 256+40, 15 );
35:    fill( 256-30, 200-30, 256+30, 200+30, 10 );
36:    *****/
37:
38:    vpage(2);
39:    SX = SY = 256;
40:    X1 = Y1 = 256;
41:    Y2 = 128;
42:    W = 128;
43:    H = 128;
44:    A = 0;
45:    a_frag = 1;
46:    XROT0_INIT();
47:    WNDROT0(0,1);
48:
49:    while(1){
50:        if( BITSNS( 0x8 ) & 0x80 ) A -= 1; /* 4 key */
51:        if( BITSNS( 0x9 ) & 0x02 ) A += 1; /* 6 key */
52:
53:        if( BITSNS( 0xE ) & 0x04 ) { SX += 10; SY += 10; } /* OP1 key */
54:        if( BITSNS( 0xE ) & 0x08 ) { SX -= 10; SY -= 10; } /* OP2 key */
55:
56:        if( BITSNS( 0xA ) & 0x20 ) SX += 10; /* XF1 key */
57:        if( BITSNS( 0xA ) & 0x40 ) SX -= 10; /* XF2 key */
58:
59:        if( BITSNS( 0xA ) & 0x80 ) SY += 10; /* XF3 key */
60:        if( BITSNS( 0xB ) & 0x01 ) SY -= 10; /* XF4 key */
61:
62:        if( BITSNS( 0x7 ) & 0x08 ) X1 -= 5; /* LEFT key */
63:        if( BITSNS( 0x7 ) & 0x20 ) X1 += 5; /* RIGHT key */
64:
65:        if( BITSNS( 0x7 ) & 0x10 ) Y1 -= 5; /* UP key */
66:        if( BITSNS( 0x7 ) & 0x40 ) Y1 += 5; /* DOWN key */
67:
68:        if( BITSNS( 0x0 ) & 0x02 ) break; /* ESC key */
69:
70:    #asm
71:
72:    VDISP: BTST.B #4,$E88001 /* 帰線待ち */
```


です。偉ぶった文章ですがそういうつもりはありません。

なんか悲観的になりましたが「このプログラムでは無理だ」という話です。ふたたび誤解のないようお願いします。もちろん僕はBG回転に挑戦するつもりです(図5)。

BG回転機能を持ったハードを販売しているのはいまのところナムコだけと思われます。そのアーケードマシンでもロクにない機能を家庭用ゲームマシンに持ち込もうというのだから、(よくも悪くも)いかにとんでもないことをスーパーファミコンがやろうとしているかがわかると思います。コストが下げられなくて当然、発売が延びて当たり前といえますね。

* * *

XROT0ということはXROT1があるだろうと容易に想像がつくわけですね。XROT1はXROT0の20%の処理速度向上を果たしたものの、画面のサイズが128ドットまでに限られるのと仮想画面の使用を強要されるという、面白くない副作用がぞくぞくと発生したので、発表はXROT0にさせていただきました。XROT1が出ないのなら識別のために「XROT」に「0」をつけておく必要はないのですが、これは単に僕の気持ちの問題です。

おしまい

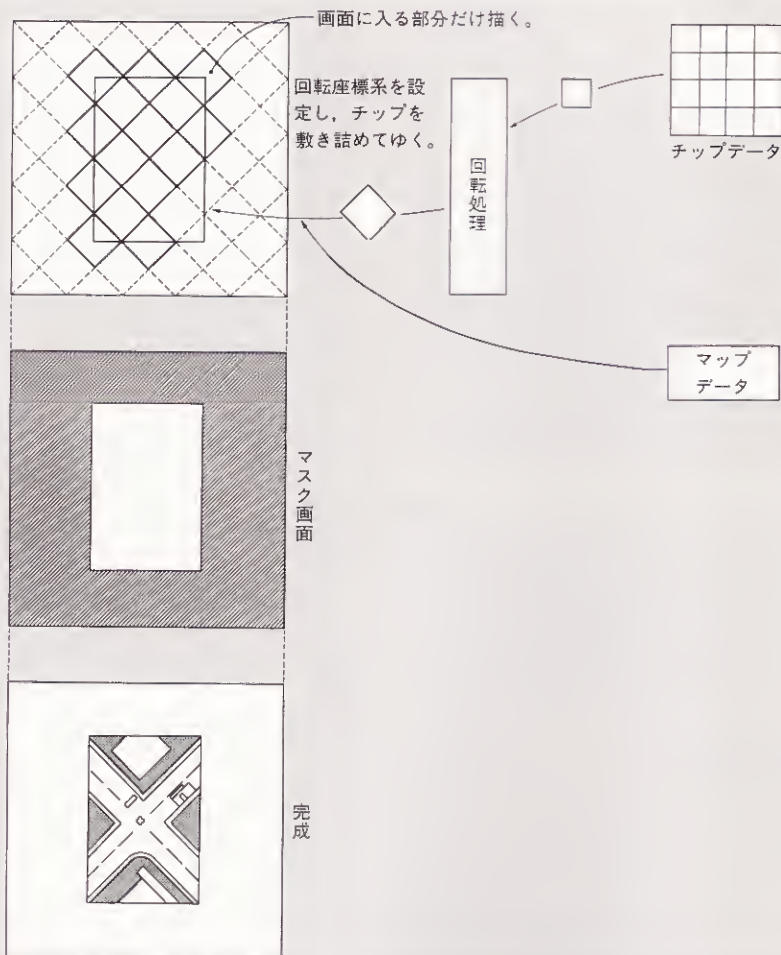
いかがでしたか？ このプログラムでX68000の限界のひとつを示したという自信があります。柴田惇氏のいい回しを借りれば「すごい」と思おうが「こんなもんか」と思おうが今後のX68000ユーザーの指針になることは確かである、というところでしょうか。あと僕としては「その筋」は死語になってはしくないので、みんなで盛り返しましょう。いろいろ偉そうなことを書いてきましたが、僕はOh!X誌上ではあまりでしゃばれないような気がします。いまさら遅いか。ではさようなら。

```

73:                                BNE.B    VDISP
74:
75:                                #endasm
76:
77:                                if( a_frag == 1 ) { X2 = 128          ; home(1,256,0); }
78:                                else { X2 = 128 + 256; home(1,000,0); }
79:
80:                                a_frag *= -1;
81:
82:                                XROT0( X1, Y1, X2, Y2, W, H, SX, SY, A );
83:
84:                                }
85:                                CRTMOD( 8 + 0x100 );
86:                                C_CURON();VPAGE(1);
87:                                B_SUPER( SSP );
88: }

```

図5 回転BGシステム



リスト3

```

===== XROT0.S =====
1: .GLOBL _XROT0.GO
2: .GLOBL _XROT0.INIT
3: .GLOBL _WNDROT0
4:
5: .TEXT
6:
7: ***** ROT 0 *****
8: * S V M O D E *****
9: _XROT0:
10:  MOVEQ.L D0-D7/A0-A4,REGBUF
11:  MOVEA.L #0,A1
12:  TRAP    #15
13:  MOVE.L D0,SSPBUF
14:  MOVE.L USP,A0
15:  MOVE.L A0,USPBUF
16:
17: * ERROR CHECK *****
18:  AND.W  #003FF,30(SP)
19:  AND.W  #003FF,34(SP)
20:  CMPI.W #128,26(SP)
21:  BHI.W  BAD_END
22:
23:  MOVE.W 22(SP),D0
24:  BEQ.W  BAD_END
25:  CMPI.W #128,D0
26:  BHI.W  BAD_END
27:  CMPI.W NOW_WIDE,D0
28:  BEQ.B  GO
29:  MOVE.W D0,NOW_WIDE
30:
31: * 書き換えサブルーチン *****
32:  LEA.L  W_LINE,A0
33:  MOVE.W 22(SP),D0
34:  ASL.W  #1,D0
35:  SUBQ.W #1,D0
36:  LOOP00:
37:  MOVE.W #034D1,(A0)+
38:  MOVE.L #043E9_0000,(A0)+
39:  DEBR.A D0,LOOP00
40:  MOVE.W #04E75,(A0)
41:
42: * 角度取り出し *****
43:  GO:
44:  MOVE.L 35(SP),D0
45:  MOVEQ.L #020,D0
46:  DIVS.W D0,D0
47:
48:  * 2F W = 0 THEN ERROR
49:  * IF 128 < W THEN ERROR
50:  * W は前回の設定値と同じか
51:  * 同じなら書き換えしない。
52:  * 書き換えサブルーチン *****
53:  * 35 W LINE データを W に設定
54:  * 35 W LINE データを W に設定
55:  * 35 W LINE データを W に設定
56:  * 35 W LINE データを W に設定
57:  * 35 W LINE データを W に設定
58:  * 35 W LINE データを W に設定
59:  * 35 W LINE データを W に設定
60:  * 35 W LINE データを W に設定
61:  * 35 W LINE データを W に設定
62:  * 35 W LINE データを W に設定
63:  * 35 W LINE データを W に設定
64:  * 35 W LINE データを W に設定
65:  * 35 W LINE データを W に設定
66:  * 35 W LINE データを W に設定
67:  * 35 W LINE データを W に設定
68:  * 35 W LINE データを W に設定
69:  * 35 W LINE データを W に設定
70:  * 35 W LINE データを W に設定
71:  * 35 W LINE データを W に設定
72:  * 35 W LINE データを W に設定
73:  * 35 W LINE データを W に設定
74:  * 35 W LINE データを W に設定
75:  * 35 W LINE データを W に設定
76:  * 35 W LINE データを W に設定
77:  * 35 W LINE データを W に設定
78:  * 35 W LINE データを W に設定
79:  * 35 W LINE データを W に設定
80:  * 35 W LINE データを W に設定
81:  * 35 W LINE データを W に設定
82:  * 35 W LINE データを W に設定
83:  * 35 W LINE データを W に設定
84:  * 35 W LINE データを W に設定
85:  * 35 W LINE データを W に設定
86:  * 35 W LINE データを W に設定
87:  * 35 W LINE データを W に設定
88:  * 35 W LINE データを W に設定
89:  * 35 W LINE データを W に設定
90:  * 35 W LINE データを W に設定
91:  * 35 W LINE データを W に設定
92:  * 35 W LINE データを W に設定
93:  * 35 W LINE データを W に設定
94:  * 35 W LINE データを W に設定
95:  * 35 W LINE データを W に設定
96:  * 35 W LINE データを W に設定
97:  * 35 W LINE データを W に設定
98:  * 35 W LINE データを W に設定
99:  * 35 W LINE データを W に設定
100:  * 35 W LINE データを W に設定

```



```

44: SWAP.W D0 *
45: EXT.L D0 *
46: ADD.L D1,D0 * D0 = 0 ... 239
47: DIVU.W D1,D0 * D0 / アマツ=0 ... 119
48: SWAP.W D0 *
49: EXT.L D0 *
50: MOVE.L D0,D5 *
51:
52: * 転送側画像 転送開始座標*****
53: ASL.W #2,D0 *
54: LEA.L XSDAT0,A0 * 三角関数データ
55: MOVE.L {A0,D0.W},D4 *
56: MOVEQ.L #12,D5 * ASR =ヨ 1/4096 エンザン ヨウ
57: MOVE.W D4,D7 * D5 = COS
58: SWAP.W D4 * D4 = SIN
59:
60: MOVE.W 30(SP),D0 * D0 = SCALE X
61: MULU.W 22(SP),D0 * D0 = W * SCALE X
62: ASR.L #8,D0 *
63: NEG.W D0 *
64: MOVE.W 34(SP),D1 * D1 = SCALE Y
65: MULU.W 26(SP),D1 * D1 = H * SCALE Y
66: ASR.L #8,D1 *
67: MOVE.W D5,D2 * D2 = COS
68: MOVE.W D4,D3 * D3 = SIN
69:
70: MULS.W D0,D2 * D2 = X * COS
71: MULS.W D1,D3 * D3 = Y * SIN
72: SUB.L D3,D2 * D2 = X,COS - Y,SIN
73: ASR.L D7,D2 * D2 = D2 / 4096
74:
75: MULS.W D5,D1 * D1 = Y * COS
76: MULS.W D4,D0 * D0 = X * SIN
77: ADD.L D0,D1 * D1 = Y,COS + X,SIN
78: ASR.L D7,D1 * D1 = D1 / 4096
79: MOVEQ.L #10,D7 * ASL =ヨ 1024 ハイ エンザン ヨウ
80:
81: MOVE.W 6(SP),D4 * X1
82: MOVE.L 8(SP),D5 * Y1
83: ADD.W D2,D4 *
84: SUB.L D1,D5 *
85: ASL.W #1,D4 * * 2
86: ASL.L D7,D5 * * 1024
87: MOVEA.L SAMPL,A0 *
88: ADDA.W D4,A0 *
89: ADDA.L D5,A0 * 転送開始アドレス完成
90:
91: * 合成側画像 描画開始座標*****
92: MOVEA.L PLAYPL,A2 *
93: MOVE.W 14(SP),D0 * X2
94: MOVE.L 16(SP),D1 * Y2
95: SUB.W 22(SP),D0 * W
96: SUB.L 24(SP),D1 * H
97: ASL.W #1,D0 * * 2
98: ASL.L D7,D1 * * 1024
99: ADDA.W D0,A2 *
100: ADDA.L D1,A2 * 描画開始アドレス完成
101:
102: * 使用する直線データ二本の先頭アドレス***
103: LEA.L XROTAT0,A3 *
104: MOVEA.L A3,A4 *
105: MOVE.W D0,D1 * カクト
106: ASL.W #8,D1 * ライン ノ データは 256 ハイ ト
107: ADDA.W D1,A3 * X ノウコウ ノ データ ヨミダシ アドレス
108: MOVE.L D0,D1 *
109: ADDI.W #90,D1 * 90 * 3 = 270
110: DIVU.W #120,D1 * 120 * 3 = 360
111: SWAP.W D1 *
112: ASL.W #8,D1 *
113: ADDA.W D1,A4 * Y ノウコウ ノ データ ヨミダシ アドレス
114:
115: * 「書き換えルーチン」のLEAのイミディエイト値を書き換える。*
116: * これは横方向拡大縮小処理
117: MOVE.W 30(SP),D0 * SCALE X
118: MOVE.W 22(SP),D2 * W
119: ASL.W #1,D2 *
120: MOVE.W D2,D7 *
121: SUBQ.W #1,D2 *
122: LEA.L W,LINE+4,A1 *
123: CLR.W D3 *
124: TST.B 30(SP) * IF SCALE X > 255 THEN 縮小
125: BNE.B NEXT00 *
126:
127: * 横方向拡大時の書き換え*****
128: LOOP01: CLR.W {A1} * D1 ショキカ ナシ カマフナイ
129: SUB.B D0,D1 *
130: BCC.W NEXT01 *
131: MOVE.W {A3,D3.W},{A1} * DATA READ POINTER INC
132: ADDQ.B #2,D3 *
133: NEXT01: DBRA.W D2,LOOP01 *
134: BRA.B NEXT02 *
135:
136: * 横方向縮小時の書き換え*****
137: NEXT00: MOVE.W D0,D4 *
138: LSR.W #8,D4 *
139: SUBQ.W #1,D4 *
140: CLR.W {A1} *
141: LOOP02: MOVE.W D4,D5 *
142: MOVE.W {A3,D3.W},D6 *
143: LOOP03: ADD.W D6,{A1} *
144: ADDQ.B #2,D3 * DATA READ POINTER INC
145: DBRA.W D5,LOOP03 *
146: SUB.B D0,D1 * D1 ショキカ ナシ カマフナイ
147: BCC.B NEXT03 *
148: MOVE.W {A3,D3.W},D6 *
149: ADD.W D6,{A1} *
150: ADDQ.B #2,D3 * DATA READ POINTER INC
151: NEXT03: ADDQ.W #5,A1 *
152: DBRA.W D2,LOOP02 *
153:
154: * 直線データをバッファに移し、拡大縮小処理を施しておく。*
155: * これは縦方向拡大縮小処理
156: NEXT02: MOVE.W 34(SP),D0 * SCALE Y
157: ASL.W 26(SP) * H
158: SUBQ.W #1,26(SP) *
159: LEA.L LINEY,A3 * ハッファ セントウ
160: MOVE.W 26(SP),D2 *
161: CLR.W D3 *
162: TST.B 34(SP) * IF SCALE Y > 255 THEN 縮小
163: BNE.B NEXT04 *
164:
165:
166: * 縦方向拡大時*****
167: LOOP04: CLR.W {A3} *
168: SUB.B D0,D1 * D1 ショキカ ナシ カマフナイ
169: BCC.B NEXT05 *
170: MOVE.W {A3,D3.W},{A3} *
171: ADDQ.B #2,D3 * DATA READ POINTER INC
172: NEXT05: ADDQ.W #2,A3 *
173: DBRA.W D2,LOOP04 *
174: BRA.B NEXT06 *
175:
176: * 縦方向縮小時*****
177: NEXT04: MOVE.W D0,D4 *
178: LSR.W #8,D4 *
179: SUBQ.W #1,D4 *
180: LOOP05: CLR.W {A3} *
181: MOVE.W D4,D5 *
182: LOOP06: MOVE.W {A4,D3.W},D6 *
183: ADD.W D6,{A3} *
184: ADDQ.B #2,D3 * DATA READ POINTER INC
185: DBRA.W D5,LOOP06 *
186: SUB.B D0,D1 * D1 ショキカ ナシ カマフナイ
187: BCC.B NEXT07 *
188: MOVE.W {A4,D3.W},D6 *
189: ADD.W D6,{A3} *
190: NEXT07: ADDQ.W #2,D3 * DATA READ POINTER INC
191: ADDQ.W #2,A3 *
192: DBRA.W D2,LOOP05 *
193:
194: * 合成側画像の改行値*****
195: NEXT06: LEA.L LINEY,A3 *
196: MOVE.W 26(SP),D0 * H
197: *ASR.W #1,D0 * 改道点
198: ASL.W #1,D7 * D7 = W
199: NEG.W D7 *
200: *ADDI.W #2048,D7 * 改道点
201: ADDI.W #1024,D7 * 改道点
202:
203: * 転送開始! *****
204: MOVE.L $0000,BUS_ERROR * バスエラー例外処理アドレス待避
205: MOVE.L #CANT,$0000 * オリジナル例外処理アドレス
206:
207: LOOP07: MOVEA.L A0,A1 * 描画側ライン スタートアドレス
208: BSR W,LINE * 1ライン転送
209: ADDA.W {A3}+,A0 * 転送側座標更新
210: *ADDA.W {A3}+,A0 * 転送側座標更新 (改道点)
211: ADDA.W D7,A2 * 描画側座標更新
212: DBRA.W D0,LOOP07 * 転送側ループ
213:
214: MOVE.L BUS_ERROR,$0000 * バスエラー例外処理アドレス復帰
215:
216: * S V M O D E E N D *****
217: BAD_END:
218: MOVE.L USPBUF,A0 *
219: MOVE.L A0,USP *
220: MOVEQ.L #81,D0 *
221: TST.L SSPBUF *
222: ENI NEXT08 * スーパーなのに
223: MOVEA.L SSPBUF,A1 * スーパーにしようとしていた。
224: TRAP #15 *
225: NEXT08: MOVEM.L REGBUF,D0-D7/A0-A4 *
226: RTS *
227:
228: * その場凌ぎのバスエラー例外処理 *
229: CANT: LEA 8(SP),SP *
230: *LEA #PC0000,A1 * 改道点
231: CLR.W {A2}+ *
232: RTE *
233:
234: *****X R O T O _ I N I T*****
235: *****X R O T O _ I N I T*****
236: _XROT0_INIT:
237: MOVEM.L D0-D1,-(SP) *
238: MOVE.W #0,-(SP) *
239: PEA DATFILE *
240: .DC.W $FF3D * DOS_OPEN
241: MOVE.W D0,D1 *
242:
243: MOVE.L #87800+$1E0,-(SP) *
244: PEA XROTAT0 *
245: MOVE.W D1,-(SP) *
246: .DC.W $FF3F * DOS_READ
247:
248: MOVE.W D1,-(SP) *
249: .DC.W $FF3E * DOS_CLOSE
250:
251: LEA 6*10+2(SP),SP *
252: MOVEM.L (SP)+,D0-D1 *
253: RTS *
254: *****X R O T O _ I N I T E N D*****
255: *****W N D R O T O*****
256: _WNDROT0:
257: MOVE.W 6(SP),D0 *
258: LEA.L $B80000,A0 *
259: LOOP08: ADDA.L #80000,A0 *
260: DBRA.W D0,LOOP08 *
261: MOVE.L A0,SAMPL *
262: MOVE.W 10(SP),D0 *
263: LEA.L $B80000,A0 *
264: LOOP09: ADDA.L #80000,A0 *
265: DBRA.W D0,LOOP09 *
266: MOVE.L A0,PLAYPL *
267: RTS *
268:
269: *****W N D R O T O E N D*****
270:
271: .DATA *
272: .DC.W 0 * 現在の横サイズ
273: SAMPL: #3 * 転送側画面の座標(0,0)のアドレス
274: PLAYPL: .DC.L $C0000 * 描画側画面の座標(0,0)のアドレス
275: DATFILE: .DC.B "XROTAT0",0 *
276: .BSS *
277: EVEN *
278: USPBUF: .DS.L 1 *
279: SSPBUF: .DS.L 1 *
280: BUS_ERROR: .DS.L 1 *
281: REGBUF: .DS.L 8*5 *
282: LINEY: .DS.W 256 *
283: W,LINE: .DS.W 3*256+2 * 書き換えプログラムエリア
284: XROTAT0: .DS.B $7800 * 直線データ読み込みエリア
285: XSDAT0: .DS.B $1E0 * 三角関数読み込みエリア
286: .END

```


HEART・負けるが勝ち

Ikeya Masahiko 池谷 昌彦

読者投稿によるCARD.FNCを使用したトランプゲームです。「負けるが勝ち」という副題どおり、できるだけカードを取らないようにゲームをすすめるなければいけません。なお、このプログラムの実行には1990年5月号で発表されたCARD.FNCが必要です。



私は以前より自分でカードデータを作ってゲームを作っていました。数度に分けてPUTするという方法では遅くてBASICでは使いものにならず弱っていました。その点、CARD.FNCはBASICでも十分実用になるので嬉しくなります。さっそく、これを使ってカードゲームを作ってみました。

ゲームの名前はHEARTです。「負けるが勝ち」と副題をつけたいと思います。このゲームは3人から6人用のトランプゲームです。4人でプレイするのがもっともバランスがとれるので、プレイヤーはコンピュータ3人と人間ひとりの構成にしました。



ゲームの内容

ゲームのルールを簡単に説明しましょう。まず、各プレイヤーは13枚ずつ手札を持ちます。親から順番に1枚ずつ手札を場にさらしていきます。このとき出せるのは親が出した台札と同じスート（記号）のカードだけです。どうしても出せない場合はなに出してもかまいません。

カードの順位は、2、3、4、5、6、7、8、9、10、J、Q、K、Aの順に強くなっていきます（ただし違うスートのものはもっとも弱い）。1巡した時点で最も強いカードを出した人が次の親になります。親は取った場札を自分のところに寄せます（手札には加えない）。このなかにはハートのカードがあったら、1枚につき1点のペナルティとなります。

手札がなくなった時点でペナルティの計算をし、もっとも少ない人が次のゲームの



最初の親になります。

1枚もハートを取らないことをクリアとします。クリアのときはほかの人のペナルティ分13点がもらえます。クリアが2人のときは、6点ずつで余った1点は次回に持ち越されます。

要するにハートのカードを取らないようにすればいいわけですが、ひとりでハートを13枚集めた場合だけは例外で、ほかの人から4点ずつもらえます。

だいたい感じがわかったでしょうか？



入力方法

CARD.FNCは1990年5月号で発表されたX68000用カードゲーム支援ツールです。6月号のディスクにも入っていたので、解凍して使ってください。以下にCARD.FNCをBASICに組み込むまでの手順を示します。

まず、ディスクを解凍します。6月号のオマケディスクをBドライブに入れた場合なら、

A>LH-E B:GAMES
とすると、GAMES.LZHに入ったデータが



Aドライブ上に展開されます。

ここで、BASICからMAKE.BASを実行すると自動的にCARD.FNCというファイルを作成します。できあがったCARD.FNCはBASICが入っているディレクトリに入れてください。

次に、

A>ED A:¥BASIC2¥BASIC.CNF
のように、エディタでコンフィギュレーションファイルを読み込みます。いちばん下の行に、

FUNC=CARD

と書き加え、ESC・Eでセーブします。これでCARD.FNCが組み込まれました。

次にBASICを起動してゲーム本体を打ち込みます。全部打ち込んだらRUNでゲームを始めてください。

* * *

プログラムのシャッフル部分はCARD.FNCのサンプルである“99”からルーチンを探しました。このルーチンは私が使っていたものよりもよく混ざるようです。

今後もCARD.FNCを使ったトランプゲームを作っていきたいと思いますので、皆さんよろしく願います。

リスト1

```
10 /*
20 /* HEART
30 /*      programmed by M.I., May22'90
40 /*
50 screen 1,1,1,1:console,,0
60 int jj,b1,b2,b3,b4,bb,m,f=0,rd=1
70 dim int cc(51),c(3,12),pp(51),p(3,12),gg(3)
80 dim int h(12),b(3),mai(3),kei(3),kuri(4),ten(4,3)
90 dim str nam(3)={"太郎","次郎","花子","貴方"}
100 palet(1,0)
110 /* main program
120 while f<>1
```

```
130 scrn()
140 play()
150 jd3()
160 endwhile
170 owari()
180 end
190 /* screen
200 func scrn()
210 int i
220 apage(3):vpage(15)
230 fill(0,0,511,511,3)
240 apage(2)
```



```

250 box(0,0,511,511,15):box(1,1,510,510,15)
260 line(2,144,509,144,15,&HFFFF)
270 line(160,2,160,143,15,&HFFFF)
280 line(320,2,320,143,15,&HFFFF)
290 line(2,384,509,384,15,&HFFFF)
300 line(280,145,280,383,15,&HFFFF)
310 for i=0 to 4
320   line(321,i*24+24,509,i*24+24,15,&HFFFF)
330 next
340 for i=0 to 5
350   line(i*30+360,25,i*30+360,143,15,&HFFFF)
360 next
370 symbol(26,42,"H E A R T",1,4,1,1,0)
380 symbol(24,40,"H E A R T",1,4,1,13,0)
390 symbol(376,6,"SCORE",2,1,1,15,0)
400 for i=0 to 3:symbol(324,i*24+53,nam(i),1,1,1,15,0):next
410 for i=1 to 4:symbol(i*30+341,29,str$(i),1,1,1,15,0):next
420 symbol(481,29,"合計",1,1,1,15,0)
430 symbol(348,154,"各自のハート",1,1,1,15,0)
440 for i=0 to 3
450   symbol((i mod 2)*112+286,(i \ 2)*104+204,nam(i),1,1,1,
15,0)
460 next
470 symbol(8,428,nam(3),1,1,1,15,0)
480 for i=0 to 3:kei(i)=0:next
490 for i=0 to 4:kuri(i)=0:next
500 endfunc
510 /* play
520 func play()
530   while rd<5
540     prep()
550     splay()
560     jd2()
570   endwhile
580 endfunc
590 /*
600 func prep()
610   int i,j,a,b,k,s
620   /* music data set
630   if rd=1 then {
640     m_init()
650     for i=1 to 8:m_alloc(i,2000):m_assign(i,i):next
660     m_trk(1,"q8023v10c3t180132 e")
670     m_trk(2,"q8023v10c3t180132 c")
680     m_trk(3,"q8032v10c2t10014 a")
690     m_trk(4,"q5056v10c5t100116aeee")
700     m_trk(5,"q7056v10c5t 8014 a")
710     m_trk(6,"q8056v10c6t 5012 c")
720     m_trk(7,"q601 v10c4t180c#8112dec#dec#dec#d8e8fgefgefge
f8g8ab-gab-gab-ga4")
730     m_trk(8,"q7019v10c4t 55b8.b16<d4>a8.b16<c>b8b8116agf#g
a4d4b8.b16<d4>a8.b16<c4>b8b8116ab<c>ag4")
740   }
750   /* deal
760   apage(1)
770   fill((rd-1)*30+331,25,(rd-1)*30+359,47,0)
780   fill(rd*30+331,25,rd*30+359,47,5)
790   for i=1 to 4:symbol(i*30+341,29,str$(i),1,1,1,15,0):next
800   if rd=1 then {
810     symbol(184,40,"ルールの説明は",1,1,1,15,0)
820     s=sel(176,96,1,1):if s=1 then rule()
830   }
840   randomize(val(mid$(time$,4,2)+right$(time$,2)))
850   for i=0 to 51:cc(i)=i+1:next
860   for i=0 to 3:mai(i)=0:next:m=13
870   if s=2 or rd=1 then {
880     er_upms()
890     symbol(200,24,"シャッフル",1,1,1,15,0)
900     symbol(224,56,"及び",1,1,1,15,0)
910     symbol(208,88,"カード配布",1,1,1,15,0)
920   }
930   fill(40,168,240,200,5)
940   symbol(64,176,"ちょっと待って下さい",1,1,1,15,0)
950   for i=0 to 12:h(i)=0:next
960   for i=0 to 99
970     a=int(rnd()*52):b=int(rnd()*52)
980     k=cc(a):cc(a)=cc(b):cc(b)=k
990   next
1000 fill(40,168,68,200,3)
1010 for i=0 to 51
1020   if cc(i)=1 then pp(i)=13:continue
1030   if cc(i)=14 then pp(i)=26:continue
1040   if cc(i)=27 then pp(i)=39:continue
1050   if cc(i)=40 then pp(i)=52:continue
1060   pp(i)=cc(i)-1
1070 next
1080 fill(69,168,96,200,3)
1090 for i=0 to 12
1100   c(0,i)=cc(i) :p(0,i)=pp(i)
1110   c(1,i)=cc(i+13):p(1,i)=pp(i+13)
1120   c(2,i)=cc(i+26):p(2,i)=pp(i+26)
1130   c(3,i)=cc(i+39):p(3,i)=pp(i+39)
1140 next
1150 for i=0 to 11
1160   for j=i+1 to 12
1170     for k=0 to 3
1180       if p(k,i)>p(k,j) then {
1190         a=p(k,i):p(k,i)=p(k,j):p(k,j)=a
1200         a=c(k,i):c(k,i)=c(k,j):c(k,j)=a
1210       }
1220     next
1230   next
1240   fill(96,168,96+(i+1)*12,200,3)
1250 next
1260 er_upms()
1270 plcd()
1280 if s=1 then {
1290   click()
1300   apage(0):fill(0,0,511,511,0):apage(1)
1310 }
1320 mkba():htmai()
1330 /* play order
1340 if rd=1 then {
1350   symbol(184,24,"順番を決めます",1,1,1,15,0)
1360   symbol(176,56,"いい時にマウスを",1,1,1,15,0)
1370   symbol(208,88,"クリック",1,1,1,15,0)

```

```

1380 mouse(1)
1390 symbol(140,177,"が最初",1,1,1,1,0)
1400 repeat
1410   jj=int(rnd()*4)
1420   fill(96,177,128,192,15)
1430   symbol(96,177,nam(jj),1,1,1,1,0)
1440   msstat(i,j,a,b)
1450   until a<>0 or b<>0
1460   mouse(0)
1470   wait(50):er_upms()
1480 } else symbol(96,177,nam(jj)+"が最初",1,1,1,1,0):wait(60)
1490 endfunc
1500 /* play
1510 func splay()
1520   repeat
1530     b1=0:b2=0:b3=0:b4=0:bb=1
1540     ssplay()
1550     jd1()
1560     m=m-1
1570     until m=0
1580   endfunc
1590 /*
1600 func ssplay()
1610   while bb<5
1620     if jj>0 and jj<=2 then {
1630       if bb=1 then {
1640         com1():bb=2:jj=jj+1
1650         if jj=3 then you():jj=0:bb=3:continue
1660       } else if bb=2 then {
1670         com(2,b2):bb=3:jj=jj+1
1680         if jj=3 then you():jj=0:bb=4:continue
1690       } else if bb=3 then {
1700         com(3,b3):bb=4:jj=jj+1
1710         if jj=3 then you():bb=5
1720       } else if jj=3 and bb=1 then you():jj=0:bb=2:continue
1730     } else if jj=3 and bb=1 then you():jj=0:bb=2:continue
1740   endwhile
1750 endfunc
1760 /* com play as 1st player
1770 func com1()
1780   int i,is,hm=0,sm=0,bc=0
1790   dsban(jj)
1800   for i=0 to m-1
1810     if p(jj,i)>13 and p(jj,i)<27 then hm=hm+1
1820     if p(jj,i)<14 then sm=sm+1
1830   next
1840   while bc=0
1850     if hm>0 then {
1860       i=sm
1870       if p(jj,i)<17 then bc=1:break
1880       if p(jj,i)<18 and h(0)+h(1)+h(2)=1 then bc=1:break
1890       if p(jj,i)<19 and h(0)+h(1)+h(2)=2 then bc=1:break
1900       if p(jj,i)<20 and h(0)+h(1)+h(2)=0 then bc=1:break
1910       if p(jj,i)<21 and h(0)+h(1)+h(2)+h(3)=0 then bc=1:br
1920     }
1930     if p(jj,i)<22 and h(0)+h(1)+h(2)+h(3)+h(4)=0 then bc
1940     =0:break
1950     if m<=5 and m=hm then bc=1:break
1960   }
1970   repeat
1980     i=int(rnd()*m)
1990     until p(jj,i)<14 or p(jj,i)>26
2000     bc=1
2010   endwhile
2020   is=i:bacd(jj,is)
2030   bl=p(jj,is):p(jj,is)=0:c(jj,is)=0:b(jj)=b1
2040   if bl>13 and bl<27 then gg(jj)=b1:h(bl-14)=1
2050   for i=0 to m-1:cc(i)=c(jj,i):pp(i)=p(jj,i):next
2060   cdleft(is)
2070   for i=0 to m-1:c(jj,i)=cc(i):p(jj,i)=pp(i):next
2080   if jj=2 then wait(15) else wait(30)
2090 endfunc
2100 /* com play as 2nd to 4th player
2110 func com(q,id)
2120   int i,is,hm=0,sm=0,ap=0,bc=0
2130   dsban(jj)
2140   for i=0 to m-1
2150     if p(jj,i)>13 and p(jj,i)<27 then hm=hm+1
2160     if p(jj,i)<14 then sm=sm+1
2170     if (p(jj,i)-1)*13=(bl-1)*13 then ap=ap+1
2180   next
2190   while bc=0
2200     if ap>0 then {
2210       if sm>0 then is=sm+hm-1:b(jj)=0:bc=1:break else {
2220         is=m-1:b(jj)=0:bc=1:break }
2230     }
2240     if ap>0 and (bl>13 and bl<27) then {
2250       switch q
2260       case 2:is=scom2(hm,sm):break
2270       case 3:is=scom3(hm,sm):break
2280       case 4:is=scom4(hm,sm)
2290     ends
2300     if is>0 then bc=1
2310     if bc=0 and p(jj,sm)<22 then is=sm:bc=1:break
2320     if bc=0 and p(jj,sm)>21 then is=sm+hm-1:bc=1:break
2330   }
2340   if ap>0 and (bl<14 or bl>26) then {
2350     switch q
2360     case 2:is=scom2():break
2370     case 3:is=scom3():break
2380     case 4:is=scom4()
2390   ends
2400   bc=1
2410   }
2420   endwhile
2430   bacd(jj,is)
2440   id=p(jj,is):p(jj,is)=0:c(jj,is)=0
2450   if ap>0 then b(jj)=id
2460   if id>13 and id<27 then gg(jj)=id:h(id-14)=1
2470   for i=0 to m-1:cc(i)=c(jj,i):pp(i)=p(jj,i):next
2480   cdleft(is)
2490   for i=0 to m-1:c(jj,i)=cc(i):p(jj,i)=pp(i):next
2500   if jj=2 then wait(15) else wait(30)
2510 endfunc
2520 /*play you
2530 func you()

```



```

2520 int i,is,x,y,l,r,ap=0,bc=0
2530 dsban(3)
2540 if bl=0 then {
2550   for i=0 to m-1
2560     if (p(3,i)-1)*13 = (bl-1)*13 then ap=ap+1
2570   next
2580 }
2590 while bc=0
2600   symbol(176,48,"出したいカードを",1,1,1,15,0)
2610   symbol(208,84,"クリック",1,1,1,15,0)
2620   mouse(1)
2630   msarea(49,401,502,495):setmspos(64,432)
2640   repeat
2650     msstat(x,y,l,r)
2660     until l<>0 or r<>0
2670     mspos(x,y)
2680     mouse(0):er_upms()
2690     if m>9 then is=(x-48)*13 else is=(x-48)*50
2700     if is>m-1 then dame():wait(40):er_upms():continue
2710     if bl=0 and ap=0 and (p(3,is)-1)*13 <> (bl-1)*13 then {
2720       dame():wait(40):er_upms():continue
2730     } else bacd(3,is):b(3)=p(3,is):bc=1
2740     if bl=0 and ap=0 then b(3)=0
2750     switch bb
2760       case 1: bl=p(3,is):break
2770       case 2: b2=p(3,is):break
2780       case 3: b3=p(3,is):break
2790       case 4: b4=p(3,is)
2800     endswhch
2810     if p(3,is)>13 and p(3,is)<27 then gg(3)=p(3,is):h(gg(3)-14)=1
2820     p(3,is)=0:c(3,is)=0
2830     for i=0 to m-1:cc(i)=c(3,i):pp(i)=p(3,i):next
2840     cdleft(is)
2850     for i=0 to m-1:c(3,i)=cc(i):p(3,i)=pp(i):next
2860     fill(2,385,509,509,0):m=m-1
2870     plcd():m=m+1
2880   endwhile
2890 endfunc
2900 /* judge1
2910 func jdl()
2920   int i,j,a
2930   dim int ba(3)
2940   for i=0 to 3:ba(i)=b(i):next
2950   for i=0 to 2
2960     for j=i+1 to 3
2970       if b(i)<b(j) then a=b(i):b(i)=b(j):b(j)=a
2980     next
2990   next
3000   for jj=0 to 3
3010     if b(0)=ba(jj) then kachi(jj):wait(40):break
3020   next
3030   a=mai(jj)
3040   for i=0 to 3
3050     if gg(i)>0 then {
3060       mai(jj)=mai(jj)+1:htcd(jj,i):htmai():gg(i)=0
3070     }
3080   next
3090   if mai(jj)>a then ha(mai(jj)-a):wait(40) else ha(0):wait(40)
3100 mkba()
3110 endfunc
3120 /* judge2
3130 func jd2()
3140   int i,j,a,b,cla
3150   m_play(6)
3160   apage(0)
3170   fill(2,145,509,383,8)
3180   box(64,208,448,352,15,&HFFFF)
3190   line(65,256,447,256,15,&HFFFF)
3200   for i=0 to 2:line(65,i*24+280,239,i*24+280,15,&HFFFF)
3210   line(313,i*24+280,375,i*24+280,15,&HFFFF)
3220   next
3230   line(120,209,120,352,15,&HFFFF)
3240   line(176,209,176,352,15,&HFFFF)
3250   line(240,209,240,352,15,&HFFFF)
3260   line(312,209,312,352,15,&HFFFF)
3270   line(376,209,376,352,15,&HFFFF)
3280   symbol(160,168,"第"+str$(rd)+"回 得点計算",1,1,2,15,0)
3290   symbol(124,217,"ハート",1,1,1,15,0)
3300   symbol(124,237,"枚 数",1,1,1,15,0)
3310   symbol(184,217,"今回の",1,1,1,15,0)
3320   symbol(184,237,"得 点",1,1,1,15,0)
3330   symbol(244,217,"前回から",1,1,1,15,0)
3340   symbol(244,237,"の 繰越点",1,1,1,15,0)
3350   symbol(320,217,"修正後",1,1,1,15,0)
3360   symbol(320,237,"の 得点",1,1,1,15,0)
3370   symbol(380,217,"次回への",1,1,1,15,0)
3380   symbol(380,237,"繰越点",1,1,1,15,0)
3390   for i=0 to 3
3400     symbol(76,i*24+261,nam(i),1,1,1,15,0)
3410     if mai(i)>9 then {
3420       symbol(132,i*24+261,str$(mai(i)),2,1,1,15,0)
3430     } else symbol(148,i*24+261,str$(mai(i)),2,1,1,15,0)
3440     if mai(i)=0 then cla=cla+1
3450   next
3460   if cla=0 then {
3470     kuri(rd)=13+kuri(rd-1)
3480     for i=0 to 3:ten(rd,i)=mai(i):next
3490   }
3500   if cla=3 then {
3510     kuri(rd)=0
3520     for i=0 to 3
3530       if mai(i)=13 then ten(rd,i)=12 else ten(rd,i)=-4
3540     next
3550   }
3560   if cla>0 and cla<3 then {
3570     kuri(rd)=13 mod cla + kuri(rd-1) mod cla
3580     for i=0 to 3
3590       if mai(i)=0 then ten(rd,i)=13*cla else ten(rd,i)=-ma
3600     next
3610     symbol(264,297,str$(kuri(rd-1)),2,1,1,15,0)
3620     symbol(398,297,str$(kuri(rd)),2,1,1,15,0)
3630     for i=0 to 3
3640       if ten(rd,i)>0 then {

```

```

3650       symbol(200,i*24+261,str$(ten(rd,i)),2,1,1,15,0)
3660     } else symbol(164,i*24+261,str$(ten(rd,i)),2,1,1,15,0)
3670   next
3680   if cla=3 then {
3690     for i=0 to 3
3700       if ten(rd,i)>0 then ten(rd,i)=ten(rd,i)+kuri(rd-1):b
3710   next
3720   }
3730   if cla=0 and cla<3 then {
3740     for i=0 to 3
3750       if ten(rd,i)>0 then ten(rd,i)=ten(rd,i)+kuri(rd-1)*c
3760   next
3770   }
3780   for i=0 to 3
3790     kei(i)=kei(i)+ten(rd,i)
3800     if ten(rd,i)>0 then {
3810       symbol(336,i*24+261,str$(ten(rd,i)),2,1,1,15,0)
3820       symbol(rd*30+342,i*24+261,str$(ten(rd,i)),1,1,1,15,0)
3830     } else {
3840       symbol(320,i*24+261,str$(ten(rd,i)),2,1,1,15,0)
3850       symbol(rd*30+342,i*24+261,str$(ten(rd,i)),1,1,1,15,0)
3860     }
3870     fill(481,i*24+48,385,i*24+71,0)
3880     if kei(i)>0 then {
3890       symbol(493,i*24+53,str$(kei(i)),1,1,1,15,0)
3900     } else symbol(485,i*24+53,str$(kei(i)),1,1,1,15,0)
3910   next
3920   rd=rd+1
3930   if rd<5 then {
3940     ten(rd-1,3)=ten(rd-1,3)+1
3950     for i=0 to 3:cc(i)=ten(rd-1,i):next
3960     for i=0 to 2
3970       for j=i+1 to 3
3980         if cc(i)<cc(j) then a=cc(i):cc(i)=cc(j):cc(j)=a
3990       next
4000     next
4010     for i=0 to 3
4020       if cc(0)=ten(rd-1,3) then jj=i:break
4030     next
4040     ten(rd-1,3)=ten(rd-1,3)+1
4050     symbol(176,48,str$(rd)+" 画面を始める",1,1,1,15,0)
4060     s=sel(176,96,2,1)
4070     if s=2 then f1:rd=5
4080   } else click()
4090   apage(1):fill(0,0,511,511,0)
4100   apage(0):fill(0,144,511,511,0):er_upms():apage(1)
4110 endfunc
4120 /* judge3
4130 func jd3()
4140   int i,j,a,b
4150   if f>1 then {
4160     vpage(9)
4170     apage(0):fill(0,0,511,511,0)
4180     for i=0 to 5
4190       box(48+i*5,88-i*5,464-i*5,432-i*5,15)
4200     next
4210     fill(79,111,431,401,2)
4220     kei(3)=kei(3)+1
4230     for i=0 to 3:cc(i)=sel(i):next
4240     for i=0 to 2
4250       for j=i+1 to 3
4260         if cc(i)<cc(j) then a=cc(i):cc(i)=cc(j):cc(j)=a
4270       next
4280     next
4290     for i=0 to 3
4300       if cc(0)=sel(i) then jj=i:break
4310     next
4320     symbol(97,218,nam(jj)+" の 得点 ",2,2,2,5,0):m_play(7)
4330     symbol(352,448,"もう一度やりませんか",1,1,1,15,0)
4340     s=sel(360,465,2,1)
4350     if s=1 then {
4360       rd=1:fill(0,0,511,511,0)
4370       apage(1):fill(0,0,511,511,0)
4380       apage(2):fill(0,0,511,511,0)
4390       vpage(15)
4400     } else f=1
4410   }
4420 endfunc
4430 /* ovari
4440 func ovari()
4450   vpage(2):apage(1)
4460   fill(0,0,511,511,2)
4470   symbol(272,400,"お宝を獲りました",1,1,2,15,0)
4480   m_play(8)
4490 endfunc
4500 /*
4510 func scom2(hm,sm)
4520   int i,is,bc=0
4530   for i=0 to sm-1
4540     is=sm-hm-1-i
4550     if p(jj,is)<b1 then bc=1:break
4560   next
4570   if bc=0 then is=0
4580   return(is)
4590 endfunc
4600 /*
4610 func scom3(hm,sm)
4620   int i,is,bc=0
4630   for i=0 to hm-1
4640     is=sm-hm-1-i
4650     if p(jj,is)<b1 or p(jj,is)<b2 then bc=1:break
4660   next
4670   if bc=0 then is=0
4680   return(is)
4690 endfunc
4700 /*
4710 func scom4(hm,sm)
4720   int i,is,bc=0
4730   for i=0 to hm-1
4740     is=sm-hm-1-i
4750     if p(jj,is)<b1 or p(jj,is)<b2 or p(jj,is)<b3 then bc=1
4760     :break
4770   next
4780   if bc=0 then is=0

```



```

4780 return(is)
4790 endfunc
4800 /*
4810 func sscom2()
4820 int i,is
4830 for i=0 to m-1
4840 is=m-1-i
4850 if (p(jj,is)-1)*13=(b1-1)*13 then break
4860 next
4870 return(is)
4880 endfunc
4890 /*
4900 func sscom3()
4910 int i,is,a=0
4920 if b2>13 and b2<27 then {
4930 for i=0 to m-1
4940 is=m-1-i
4950 if (p(jj,is)-1)*13=(b1-1)*13 and p(jj,is)<b1 then a=
1:break
4960 next
4970 if a=0 then {
4980 for i=0 to m-1
4990 is=m-1-i
5000 if (p(jj,is)-1)*13=(b1-1)*13 then break
5010 next
5020 }
5030 } else is=sscom2()
5040 return(is)
5050 endfunc
5060 /*
5070 func sscom4()
5080 int i,is,a=0
5090 if (b2>13 and b2<27) and (b3>13 and b3<27) then {
5100 for i=0 to m-1
5110 is=m-1-i
5120 if (p(jj,is)-1)*13=(b1-1)*13 and p(jj,is)<b1 then a=
1:break
5130 next
5140 if a=0 then {
5150 for i=0 to m-1
5160 is=m-1-i
5170 if (p(jj,is)-1)*13=(b1-1)*13 then break
5180 next
5190 }
5200 } else if (b2>13 and b2<27) and (b3<14 or b3>26) then {
5210 for i=0 to m-1
5220 is=m-1-i
5230 if (p(jj,is)-1)*13=(b1-1)*13 and (p(jj,is)<b1 or p(j
j,is)<b3) then a=1:break
5240 next
5250 if a=0 then {
5260 for i=0 to m-1
5270 is=m-1-i
5280 if (p(jj,is)-1)*13=(b1-1)*13 then break
5290 next
5300 }
5310 } else if (b2<14 or b2>26) and (b3>13 and b3<27) then {
5320 for i=0 to m-1
5330 is=m-1-i
5340 if (p(jj,is)-1)*13=(b1-1)*13 and (p(jj,is)<b1 or p(j
j,is)<b2) then a=1:break
5350 next
5360 if a=0 then {
5370 for i=0 to m-1
5380 is=m-1-i
5390 if (p(jj,is)-1)*13=(b1-1)*13 then break
5400 next
5410 }
5420 } else is=sscom2()
5430 return(is)
5440 endfunc
5450 /*
5460 func cleft(k)
5470 int i
5480 for i=0 to m-k:cc(k+i)=cc(k+i+1):pp(k+i)=pp(k+i+1):next
5490 endfunc
5500 /*
5510 func sel(x,y,m,n)
5520 int i,j,a,b
5530 str mm,nn
5540 switch m
5550 case 1:mm=" 必 要":break
5560 case 2:mm=" O K"
5570 endswitch
5580 switch n
5590 case 1:nn=" 不 要":break
5600 case 2:nn=" やめる"
5610 endswitch
5620 fill(x,y,x+56,y+24,15):fill(x+72,y,x+128,y+24,15)
5630 symbol(x+4,y+4,mm,1,1,1,15,0)
5640 symbol(x+76,y+4,nn,1,1,1,1,0)
5650 mouse(1)
5660 msarea(x+1,y+1,x+127,y+23)
5670 setmpos(x+28,y+8)
5680 repeat
5690 msstat(i,j,a,b)
5700 until a<>0 or b<>0
5710 mpos(i,j)
5720 mouse(0)
5730 if i<x+64 then {
5740 fill(x,y,x+56,y+24,1):symbol(x+4,y+4,mm,i,1,1,15,0):s=1
5750 symbol(x+4,y+4,mm,1,1,1,15,0):s=1
5760 } else {
5770 fill(x+72,y,x+128,y+24,1)
5780 symbol(x+76,y+4,nn,i,1,1,1,15,0):s=2
5790 }
5800 return(s):wait(40)
5810 endfunc
5820 /*
5830 func click()
5840 int i,j,a,b
5850 symbol(176,48,"よければマウスを",1,1,1,15,0)
5860 symbol(208,84,"クリック",1,1,1,15,0)
5870 mouse(1)

```

```

5880 msarea(176,48,288,96)
5890 setmpos(232,70)
5900 repeat
5910 msstat(i,j,a,b)
5920 until a<>0 or b<>0
5930 mouse(0)
5940 er_upms()
5950 endfunc
5960 /*
5970 func mkba()
5980 fill(3,145,279,383,8):fill(40,168,240,200,15)
5990 for i=0 to 3:symbol(i*56+40,344,nam(i),1,1,1,15,0):next
6000 endfunc
6010 /*
6020 func dsban(j)
6030 er_ms():symbol(108,177,nam(j)+"の番",1,1,1,1,0)
6040 endfunc
6050 /*
6060 func kachi(j)
6070 er_ms():symbol(48,177,nam(j)+"の勝ち",1,1,1,1,0)
6080 endfunc
6090 /*
6100 func ha(ht)
6110 symbol(134,177,"ハート "+str$(ht)+"枚",1,1,1,1,0):m_play
(4)
6120 endfunc
6130 /*
6140 func ha0()
6150 symbol(134,177,"ハート無し",1,1,1,1,0):m_play(5)
6160 endfunc
6170 /*
6180 func dame()
6190 er_upms()
6200 symbol(200,48,"出せません",1,1,1,1,0):m_play(3)
6210 endfunc
6220 /*
6230 func bacd(j,i)
6240 c_put(j*56+32,225,c(j,i)):m_play(1,2)
6250 endfunc
6260 /*
6270 func plcd()
6280 int i
6290 if m>9 then {
6300 for i=0 to m-1
6310 c_put(i*34+48,400,c(3,i))
6320 line(i*34+47,400,i*34+47,496,1)
6330 m_play(1,2)
6340 next
6350 } else {
6360 for i=0 to m-1
6370 c_put(i*50+48,400,c(3,i))
6380 m_play(1,2)
6390 next
6400 }
6410 symbol(16,453,str$(m),1,1,1,15,0)
6420 endfunc
6430 /*
6440 func htod(jj,a)
6450 int a,b,h
6460 a=(jj mod 2)*112+317:b=(jj ¥ 2)*104+176
6470 if gg(a)=26 then h=14 else h=gg(a)+1
6480 c_put(a+mai(jj)*2,b,h)
6490 line(a+mai(jj)*2-1,b,a+mai(jj)*2-1,b+96,1)
6500 m_play(1,2)
6510 endfunc
6520 /*
6530 func htmai()
6540 int i
6550 for i=0 to 3
6560 fill((i mod 2)*112+294,(i ¥ 2)*104+228,(i mod 2)*112+3
10,(i ¥ 2)*104+244,0)
6570 symbol((i mod 2)*112+294,(i ¥ 2)*104+228,str$(mai(i)),
1,1,1,15,0)
6580 next
6590 endfunc
6600 /*
6610 func wait(t)
6620 int i
6630 for i=0 to t*100:next
6640 endfunc
6650 /*
6660 func er_upms()
6670 fill(161,3,319,143,0)
6680 endfunc
6690 /*
6700 func er_ms()
6710 fill(40,168,240,200,15)
6720 endfunc
6730 /*
6740 func rule()
6750 apage(0)
6760 fill(2,145,509,383,8)
6770 symbol(196,160,"ルール",1,1,1,15,0)
6780 symbol(60,190,"1: カードの強さは A,K,Q,J...4,3,2 の順",
1,1,1,15,0)
6790 symbol(60,208,"2: 各自が 1 枚ずつ同じ種類を出さねばなり
ません",1,1,1,15,0)
6800 symbol(120,225,"但し手持ちが無ければ、何でもかまいません
",1,1,1,15,0)
6810 symbol(60,243,"3: 最も強い札を出した人が 4 枚全部取りま
す",1,1,1,15,0)
6820 symbol(60,261,"4: 取った中にハートがあれば 1 枚につき
1 点",1,1,1,15,0)
6830 symbol(60,279,"5: 持ち札が無くなるまで繰り返します",1,1
,1,15,0)
6840 symbol(60,297,"6: プラス点はハートを取らなかった人で分
けます",1,1,1,15,0)
6850 symbol(120,314,"1 人の時... + 13 点",1,1,1,15,0)
6860 symbol(120,331,"2 人の時... + 6 点、残りは繰り越し",
1,1,1,15,0)
6870 symbol(60,349,"7: 1 人でハート 13 枚取れば +12 点、他の
人は -4 点",1,1,1,15,0)
6880 apage(1)
6890 endfunc

```


トランジェントコマンドを作る

亀田 雅彦 Kameda Masahiko

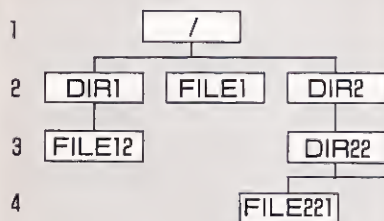
X1turbo用
ディスク管理プログラム
INTEGRAL X1

KAME-DOSをもっとDOSらしく使うための方法として、KAME-DOSの外部コマンドを作成してみましょう。ディスク管理のほか、さまざまなプログラムがコマンドとして使用できます。こういったものがBASICで記述できるのです。

先月号でノーマルX1にも対応して、いよいよ本格的になってきました。もし、X1ユーザーでまだKAME-DOSを入手していない方は、ぜひバックナンバーなどから入力するようにしてください。

6、7月号のプログラムだけではなかなかその威力を発揮しないKAME-DOSも、今月から紹介していく外部コマンドを活用すればその世界が広がります。特にノーマルX1ユーザーには、ディスク関係の命令がturboBASICに匹敵するようになるので、お楽しみに。また、外部コマンドのノウハウが蓄積していくとユーザー自身の手でKAME-DOSワールドを広げていくことができるようになります（もちろん最初の公約どおり、BASICで）。

図1 階層ディレクトリ



このように、ディレクトリの下にまた下位ディレクトリを作ってファイルの整理をしやすくする構造を階層ディレクトリといいます。1段目のことをルートディレクトリと呼んで、ここから「見える」のは、2段目だけで3段目以降は見えません。また、DIR1にいるときはルートディレクトリやDIR2とは関係なくなるので、同じファイル名を使っても上書きされません。本文中の「フルパス」というのは、ファイル名をルートディレクトリから全部表示したものです。FILE221をフルパスにする

と、

/DIR2/DIR22/FILE221
になります。ルートディレクトリは「/」（スラッシュ）です。カレントディレクトリというのは、現在自分のいるディレクトリで「/」とかDIR1とかDIR22となります。

とりあえず、今月から何回かに分けて、普通のDOSにあるような命令を外部コマンドとして発表しながら、その動作と作り方を説明しましょう。基本的に外部コマンドも内部コマンドも（COMMAND.X1内に用意されているもの）、作りは同じなのでCOMMAND.X1の理解の助けにもなると思います。

それでは、今月は「MD.X1」「RD.X1」そして、特集と関連して「GLOAD.X1」「GSAVE.X1」を発表してみましょう。

外部コマンドワールド

リスト3が「MD.X1」です。turboBASICでいうところのMKDIRにあたります。下位ディレクトリをカレントディレクトリの下に作るのですが、難しいところなので階層ディレクトリ全般について簡単に図1で説明しておきます。また、階層ディレクトリとは切っても切れない関係にあるCD（ディレクトリの変更）命令については、内部コマンドなので6月号に解説されています（でも、6月号ではちょっと手抜き解説が多かったと反省することしきりです）。

反省ばかりしていても進歩がないので、さっそく使い方に入ります。

命令：MD（MKDIR）

書式：MD 新規ディレクトリネーム

プログラム：リスト3

まずリスト3を打ち込んでください。使用BASICは、いま自分の持っているINTEGRAL.Xが動いているものならなんでも大丈夫です（CZ-8FB01ver1.0, turboBASIC,Z-BASIC）。用途別に自分でBASICシステムを構築してください。ただしCZ-8FB01では日本語入力ができないので、リストの一番最後にDATA文としてまとめら

れているメッセージは、注釈行のほうを生かして日本語のほうは打ち込まなくて結構です。たとえばリストで、

```
1650 LABEL "d1": DATA エラーが  
発生しました!!
```

```
1660 LABEL "d1": DATA Error !!  
という2行は、
```

```
1650 `  
1660 LABEL "d1": DATA Error !!
```

このようにします。これが2行ずつ組になっているので、それぞれについて変更してください。日本語入力できて、しかも使用中に日本語表示ができる（をしたい）場合には（ディスプレイの関係で表示できないこともある）、そのまま入力してください。以後、外部コマンドの入力形式はだいたい同じようなかたちになります。

使い方：

入力したら、カレントドライブかパスの通っているドライブにセーブしてください。INTEGRAL Xのコマンドライン（[X:/]の状態）から、セーブしたときのファイル名（この場合は「MD.X1」か「MKDIR.X1」）をタイプしてリターンキーを押してください。「MD」か「MKDIR」だけで、拡張子はいりません。

6月号でも書いたことですが、拡張子が「.X1」のBASICファイルはKAME-DOSの外部コマンドとして認識されます。見かけ上は、内部コマンドの実行となら変わりありません。また、コマンドラインからパラメータとして与えられる新規ディレクトリネームの書式については、囲みを参照してください。

パスが通ってなかったり、ファイル名をタイプミスしたときはエラーになります。エラーが起きずに、しばらくすると外部コマンドがロードされて起動します。指定に

間違いがなければ、下位ディレクトリを作成して、パスに従って「COMMAND.X1」をロードしなおしてコマンドラインに復帰します（CP/Mでいうリブート）。ここで、外部コマンド実行の際の注意点を挙げておきましょう。

1) 外部コマンドのファイル名は、内部コマンドのコマンド名にあたるものなのでわかりやすくすること（片仮名などにするとあとで苦労します）。拡張子は「.X1」にすること。

2) 外部コマンド実行中にブレイクして実行を強制的に中止したときは、必ず「COMMAND.X1」を実行するところから始めてください。外部コマンドをブレイクしてそのままRUNすると、変数がクリアされるのが最悪の場合暴走します。これは入力したプログラムをデバッグしているときも同じことで、エラーが出て止まったら、入力ミスを訂正していったんセーブして、「COMMAND.X1」をRUNしてそのコマンドラインから外部コマンドを実行するようにしてください（図2）。

3) 「COMMAND.X1」は必ずパスの通っているドライブにセーブしておいてください。リブートするときにパスの順に従って「COMMAND.X1」を探すので、みつからないと「リブートできません」というメッセージが出て実行が止まります。コマンドラインからの実行のときは違って、カレントドライブでもパスが通ってないと探しにいきません。

4) 外部コマンドからリブートした時点で、下位ディレクトリにいてもすべてルートディレクトリに戻されます。たとえば、[A:/TEST/] から「MD」を実行して戻ってくると、[A:/] になっているということ

ファイルネーム

ディレクトリの名前も、基本的にそのディスクフォーマットのファイル名と同じです。ファイル名の方は各マニュアルをみてもうなり、6月号にも少し解説しておきました。turboBASICの場合ディレクトリの拡張子は「.DIR」になるので、それにあわせておきました。フルパスで指定もできますし、カレントドライブからの指定もできます。図1のDIR22の下にDIR33を作りたいのなら、ルートから「MD /DIR2/DIR22/DIR33」か、DIR22から「MD DIR33」です。消したい場合は、MDをRDに変えてください。

す。これが外部コマンドと内部コマンドが見かけ上異なる唯一の点です。

上記のうち、特に2)が大切なので必ず守ってください。このほかにも外部コマンド実行中にさまざまなエラーが発生する可能性があります。その場合はエラーメッセージを出力し、実行を中止して「COMMAND.X1」へ復帰しようとしています。エラーメッセージは個々の外部コマンドが持っているものなので、統一されていません。

以上のことは、外部コマンド全般についていえることなので、これからも覚えておいてください。

命令：RD (RMDIR)

書式：RD 消去するディレクトリネーム
プログラム：リスト4

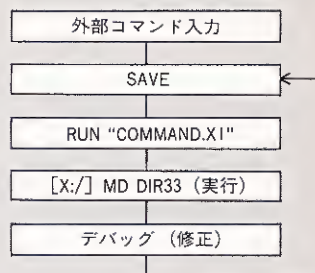
MDの逆で既存のディレクトリを消します。使用BASICも、その日本語部分の入力の仕方もMDと同じです。

使い方：

セーブする際の注意や、実行の仕方についてもMDののちを見てください。機能的にはturboBASICのものと同じです。ディレクトリ内にファイルが残っているときは、消去できません。ファイルをDELコマンドですべて消してから実行してください。

以上、2つの外部コマンドを紹介しましたが、これによって作成されたディレクト

図2 外部コマンドの入力法



上記のような手順を繰り返してください。いちいちディスクで実行していると面倒なので、G-RAMをMEM:にして実行するといいたし。COMMAND.X1やMD.X1もRAMディスクにすると速くなります。そのときも、入力ミスで暴走してディスクの破壊が起こるかもしれないので、別のディスクにプログラムを入れておきましょう。

りなどは完全にBASICとの互換性があるので、ファイルのやりとりも自由にできます。

でも、CZ-8FB01には階層ディレクトリ機能がないので、KAME-DOSで作った下位ディレクトリにはBASIC側からはアクセスできません。なお、BASICのみならずMS-DOSフォーマット（2D、2HD）とも互換性があるので、MS-DOSディスクにディレクトリを作成しようとすれば自動的にプログラム側で判断して、MS-DOSフォーマ

リスト1 GLOAD.X1

```
1000 'GLOAD.X1 Ver 1.0 By Kameda
1010 '
1020 DEFINT a-z:IF PEEK(&HD07F) THEN KLIST 0
1030 CONSOLE 0,25
1040 DEFUSR1=m_opens:DEFUSR2=m_preop
1050 '
1060 iomm=PEEK(v_iomm):baddr$=MEM$(v_baddr,2):ff$=MEM$(v_ff,2)
1070 MEM$(s_ff,2)=MKI$(&H2000)
1080 bsiz!=&HC0&H100:MEM$(v_bsiz,2)=MKI$(bsiz!):POKE v_iomm,1
1090 POKE v_dn,PEEK(s_dn):IF fe$(1)=" THEN "12"
1100 POKE &HE137,4:POKE v_mac,PEEK(s_mac4+PEEK(v_dn))
1110 IF PEEK(&HD07F)=0 THEN dirg=PEEK(&HE139):POKE &HE139,8
1120 '----- ( MAIN ROUTINE ) -----
1130 '
1140 GOSUB 1380
1150 k=PEEK(v_stop):IF k=3 THEN "13" ELSE IF k<>0 THEN "!"
1160 CLS:f$=MEM$(v_fnam+13,3):k=PEEK(&HD07F)
1170 '
1180 IF f$="GL0" OR f$="gl0" THEN IF k THEN WIDTH 40,25,0,1 ELSE WIDTH 40
1190 IF f$="GL1" OR f$="gl1" THEN IF k THEN WIDTH 80,25,0,1 ELSE WIDTH 80
1200 IF f$="GM0" OR f$="gm0" THEN IF k THEN WIDTH 40,25,0,2 ELSE WIDTH 40
1210 IF f$="GM1" OR f$="gm1" THEN IF k THEN WIDTH 80,25,0,2 ELSE WIDTH 80
1220 IF f$="GH0" OR f$="gh0" THEN IF k THEN WIDTH 40,25,1,2 ELSE WIDTH 40
1230 IF f$="GH1" OR f$="gh1" THEN IF k THEN WIDTH 80,25,1,2 ELSE WIDTH 80
1240 IF f$="GL2" OR f$="gl2" THEN IF k THEN WIDTH 40,25,0,1 ELSE WIDTH 40
1250 IF f$="GL3" OR f$="gl3" THEN IF k THEN WIDTH 40,25,0,1:OPTION SCREEN 4 ELSE
WIDTH 40
1260 INIT:IF k THEN POKE v_wfd0,PEEK(&HF8D6)
1270 GOSUB 1380:IF PEEK(v_stop) THEN "!"
1280 '
1290 POKE v_iomm,1:CALL m_dev:IF PEEK(v_stop) THEN "!"
1300 IF PEEK(v_iofg)=0 GOTO 1320
1310 POKE v_iomm,2:CALL m_dev:IF PEEK(v_stop) THEN "!"
1320 '
1330 GOSUB "ending"
1340 POKE &HE137,6:POKE v_iomm,iomm:MEM$(s_ff,2)=ff$:MEM$(v_baddr,2)=baddr$
1350 CONSOLE 0,24:IF PEEK(&HD07F) THEN KLIST 1 ELSE POKE &HE139,dirg
1360 proces=proces-1:CHAIN proces$(proces)
1370 '----- ( OPEN ) -----
1380 '
1390 MEM$(v_baddr,2)=MKI$(&H3000)
```



```

1400 POKE v_ddrv+1,7,1:POKE v_iofg,0:POKE s_escp,0:fe$=fe$(1)
1410 POKE v_od,1:d$=USR2(fe$):fe$=RIGHT$(fe$,PEEK(v_yen))
1420 IF PEEK(v_stop) RETURN
1430 POKE v_sbdr,1:POKE v_op,0:d$=USR1(fe$)
1440 MEM$(v_badr,2)=MKI$(&H4000):RETURN
1450 '-----{ END }-----
1460 '
1470 LABEL "ending"
1480 CONSOLE 0,25:CLS:CFLASH 1:PRINT "PUSH SPACE":CFLASH 0
1490 REPEAT:AS$=INKEY$:UNTIL AS$=" "
1500 CLS:RETURN
1510 '-----{ ERROR ROUTINE }-----
1520 '
1530 LABEL "!4":RESTORE "m3":GOTO 1570
1540 LABEL "!3":RESTORE "m2":GOTO 1570
1550 LABEL "!2":RESTORE "m1":GOTO 1570
1560 LABEL "!":RESTORE "m0"
1570 READ m$:BEEP:CLS:CREV 1:PRINT m$;:CREV 0:PRINT
1580 POKE v_stop,0:GOTO 1340
1590 '-----{ DATA AREA }-----
1600 LABEL "m0":DATA エラーが発生しました!!
1610 'LABEL "m0":DATA Error !!
1620 LABEL "m1":DATA ファイル・ネームを指定してください
1630 'LABEL "m1":DATA Need FILE-NAME
1640 LABEL "m2":DATA ファイルが見つかりません
1650 'LABEL "m2":DATA FILE Not Found
1660 LABEL "m3":DATA リポートできません
1670 'LABEL "m3":DATA Not REBOOT

```

ットのディレクトリを作ります。ユーザーはフォーマットの違いを意識する必要はありません。これを使えば「X68000のディスクをXlturboZで編集する」といったことも可能です。

グラフィックローダ/セーブ

グラフィック特集にあわせて、画面のロード/セーブを行うプログラムをKAME-DOS上で開発しました。特集のはうのプログラムはturboZオンリーですが、このローダとセーブはX1シリーズ全機種で使用可能です。Z-BASIC以外の標準BASICにはこのような命令がなかったので、X1間での画像のやりとりも多少便利になると思います。詳しい説明は特集記事に譲るので、ここではその紹介だけしておきましょう。

命令: GLOAD

書式: GLOAD ファイルネーム

プログラム: 特集を参照

使い方はほかの外部コマンドと同じです。特集のプログラムから子プロセス的に呼び出されるので、通常の外部コマンドとは少し異なります。

命令: GSAVE

書式: GSAVE ファイルネーム

プログラム: 特集を参照

GLOADと同じ。

リスト2 GSAVE.X1

```

1000 'GSAVE.X1 Ver 1.0 By Kameda
1010 '
1020 DEFINT a-z:INIT:IF PEEK(&HD07F) THEN KLIST 0
1030 CONSOLE 0,25
1040 DEFUSR1=m_opens:DEFUSR2=m_preop
1050 '
1060 CLS:LOCATE 10,10:PRINT "SAVE GRAM= [1] 96K"
1070 LOCATE 10,12:PRINT " [2] 64K":COLOR 5
1080 LOCATE 10,14:PRINT "[space]=GRAPHIC ON OFF":COLOR 7
1090 LOCATE 10,16:PRINT " PUSH [1] or [2]";
1100 k=0:REPEAT:as$=INKEY$(1):sx=VAL(as)
1110 IF as$=" " THEN IF k=0 THEN k=1:SCREEN ELSE k=0:PALET
1120 UNTIL 1<sx AND sx<2:PRINT sx
1130 '
1140 iomm=PEEK(v_iomm):badr$=MEM$(v_badr,2):ff$=MEM$(v_ff,2)
1150 MEM$(s_ff,2)=MKI$(&H1800):MEM$(v_badr,2)=MKI$(&H3000)
1160 IF sx=1 THEN bsiz:=&HC0*H100 ELSE bsiz:=&H60*H100
1170 MEM$(v_bsiz,2)=MKI$(bsiz!):POKE v_iomm,1
1180 POKE v_dn,PEEK(s_dn):IF fe$(1)=" " THEN "12"
1190 POKE &HE137,4:POKE v_mac,PEEK(s_mac4+PEEK(v_dn))
1200 IF PEEK(&HD07F)=0 THEN dir$=PEEK(&HE139):POKE &HE139,8
1210 '-----{ MAIN ROUTINE }-----
1220 '
1230 GOSUB 1520:IF PEEK(v_stop)<>0 THEN ""
1240 MEM$(v_badr,2)=MKI$(&H4000)
1250 '
1260 i=1:k=15:IF m=2 THEN k=1 ELSE IF m=4 THEN k=2
1270 IF sx=2 THEN 1330
1280 POKE v_iomm,1:POKE v_od,2:POKE v_iofg,2:POKE v_edr,0:CALL m_devi
1290 IF PEEK(v_stop) THEN "!"
1300 POKE v_iomm,2:POKE v_od,2:POKE v_iofg,2:POKE v_edr,k:CALL m_devi
1310 IF PEEK(v_stop) THEN "!"
1320 fx$=MKI$(&H2000):fm$=MKI$(&H8000)+MKI$(1):GOTO 1390
1330 '
1340 POKE v_iomm,1:POKE v_od,2:POKE v_iofg,2:POKE v_edr,0:CALL m_devi
1350 IF PEEK(v_stop) THEN "!" ELSE MEM$(v_badr,2)=MKI$(&HA000)
1360 POKE v_od,2:POKE v_iofg,2:POKE v_edr,k:CALL m_devi
1370 IF PEEK(v_stop) THEN "!"
1380 fx$=MKI$(&HC000):fm$=fx$+MKI$(0)
1390 '
1400 z=1:f$=fx$+MKI$(0):IF m=2 OR m=4 THEN z=0:f$=fm$
1410 POKE v_zoku+2,z:MEM$(v_fszl,4)=f$
1420 POKE v_od,2:MEM$(v_badr,2)=MKI$(&H3000):CALL m_saved:CLS
1430 IF PEEK(v_stop) THEN "!"
1440 '
1450 CFLASH 1:PRINT "PUSH ANY KEY":CFLASH 0
1460 REPEAT:AS$=INKEY$:UNTIL AS$<>" ":CLS:POKE &HE137,6
1470 IF PEEK(&HD07F)=0 THEN POKE &HE139,16
1480 POKE v_iomm,iomm:MEM$(s_ff,2)=ff$:MEM$(v_badr,2)=badr$
1490 CONSOLE 0,24:IF PEEK(&HD07F) THEN KLIST 1 ELSE POKE &HE139,dir$
1500 proces=proces-1:CHAIN proces(process)
1510 '-----{ OPEN }-----
1520 '
1530 POKE v_ddrv+1,1,7:POKE v_iofg,0:POKE s_escp,0:fe$=fe$(1)
1540 POKE v_od,2:d$=USR2(fe$):fe$=RIGHT$(fe$,PEEK(v_yen))
1550 m=PEEK(v_mac):IF PEEK(v_stop) RETURN
1560 POKE v_sbdr,1:POKE v_op,3:d$=USR1(fe$):RETURN
1570 '-----{ ERROR ROUTINE }-----
1580 '
1590 LABEL "!3":RESTORE "m2":GOTO 1620
1600 LABEL "!2":RESTORE "m1":GOTO 1620
1610 LABEL "!":RESTORE "m0"
1620 READ m$:BEEP:CLS:CREV 1:PRINT m$;:CREV 0:PRINT
1630 POKE v_stop,0:GOTO 1440
1640 '-----{ DATA AREA }-----
1650 LABEL "m0":DATA エラーが発生しました!!
1660 'LABEL "m0":DATA Error !!
1670 LABEL "m1":DATA ファイル・ネームを指定してください
1680 'LABEL "m1":DATA Need FILE-NAME
1690 LABEL "m2":DATA リポートできません
1700 'LABEL "m2":DATA Not REBOOT

```

解説! プログラミング

今月は短くて、しかもBASICプログラムなので難しいことはありません。ですから「外部コマンドの作成作法について」を中心に展開してみましょう。

第1部 起動

まず、コマンドラインからコマンド名が打ち込まれました。COMMAND.X1はそれが内部コマンドではないと判断して、ドライブにコマンドと同じファイル名を探しにいきます。なければエラーで、あれば拡張子が「.X1」かどうか(外部コマンドかどうか)を見て処理を振り分けます(図3)。

外部コマンドならCHAINして、そうじゃなければRUNします。ここが重要で、CHAINによってCOMMAND.X1で定義された変数とそのまま引き継がれます。外部コマンド側では必要に応じてその変数を

図3 外部コマンドの動作



使うことになります。だから、実行中にプログラムを止めて再実行することができないのです。これは必要な変数を何度も定義しないようにして、外部コマンド側の負担を軽くするためです。

それならば、ここでいう必要な変数とはなんでしょう？ 内部・外部に関わらずコマンドを実行するときには、KAME-DOS共通のD000番地以降に常駐しているマシン語プログラムをアクセスします（7月号のアセンブルリストのこと）。マシン語オンリーで開発しているのならアドレスはラベルに固定できますが、BASICによる開発だとアドレスを変数に定義して、ラベルとして使う必要があります。いわばこれらはグローバル変数で、コマンド内でのみ使われるのがローカル変数というところです。なお、COMMAND.X1をリポートすると一度すべての変数をクリアするので、使用変数がたまりすぎることはありません。

第2部 実行

外部コマンドはその利用目的によって相当異なった作りになるので、一言ではいきれないものがあります。ただ、大きく分けると次の3つになります。

- 1) ファイルを扱うコマンド
- 2) ディスクを扱うコマンド
- 3) それ以外のコマンド

1)は、主に内部コマンドに採用されているものでCOPYやDIRなどになります（もちろんCOPYと同じことをする外部コマンドを作ることも可能です）。特徴として、ファイルをアクセスする前には必ずそのファ

イルをOPENし、書き込んだあとにはCLOSEするということです。そのためOPEN/CLOSEルーチン呼び出す必要があります。また、実際にファイルの中身をアクセスするルーチンも使われるでしょう。開発する場合は、一番面倒なコマンドになります。

2)はFORMATやDISKCOPYのことです（今は発表できませんが、そのうちに発表したいと思います）。これらはディレクトリとかFATとか、ランダムアクセスの部分がいらないので比較的簡単に開発できます。マシン語ルーチンも低級な（ハードを直接アクセスする）ルーチンを使うのでわかりやすくなります。

3)は、特にKAME-DOS上で開発する必要はないようなプログラムです。ご存じのとおりKAME-DOSはディスクアクセスルーチンの集合体です。そのうえでディスクを使わないようなプログラムを動かしても、マシン語の常駐部分だけメモリの無駄になります。開発環境も整備されていないので、このようなプログラムは発表しないつもりです。

今月のMD,RD,GLOAD,GSAVEは1)にあたります。MD,RDの解説で、外部コマンドの雰囲気をつかんでください。

第3部 リポート

COMMAND.X1へリポートするときには、それ専用のマシン語ルーチンが用意されています。注意点は前に書いてあるとおりです。GLOAD,GSAVEに関してはこのルーチンを使わずに単なるCHAIN命令で

済ませていますが、普通はリポートルーチンを使います。最終的にはCHAINを使うので、COMMAND.X1に戻ったときの結果は同じです。

これは別にCOMMAND.X1へのリポートだけじゃなくて、「PROCESS\$」という変数で管理されているひとつ上の親プロセスへ戻るために使います。つまり、COMMAND.X1というのは親プロセスというかたちになっています。

MDとRD

どちらのプログラムでもまず、DEFUSRを定義しています。これらは先に解説したマシン語ルーチンのアドレスです。変数の頭に「M,V,S」がついているのはCOMMAND.X1からの持ち越し変数です。次に、いわゆる「OPEN」と「ディレクトリ名が指定してあるかどうか」のチェックをしま

MS-DOSのディレクトリ

MS-DOSフォーマットのディレクトリ管理方法は、X1のそれとちょっと違っています。下位ディレクトリの先頭には、「.」「..」というファイル名が2つ記録されています。これはMS-DOSでは「カレントディレクトリ」と「親ディレクトリ」を表していて、そのクラスタ番号も記録してあります。実際にこれらを使って管理しているかどうかはわかりませんが、X1にはこれに相当するものがあります。そこで、KAME-DOSでは上記のようなファイル名が出てきたら無視を決め込みます。親ディレクトリのクラスタ番号は、内部ワークエリアに保存しておくようにしました。

リスト3 MD.X1

```

1000 'MD (MKDIR) ver 1.0 By M.Kameda
1010 '
1020 DEFUSR0=m_opens:DEFUSR1=m_preop:DEFUSR2=&HEE80:DEFUSR3=m_tranc
1030 '
1040 POKE v_dn,PEEK(s_dn):POKE v_mac,PEEK(s_mac4+PEEK(s_dn))
1050 POKE v_od,1:GOSUB 1260:IF PEEK(v_stop) THEN "erre"
1060 IF fe$(1)="" OR fe$(1)="/" THEN "errx"
1070 ON PEEK(v_mac) GOSUB 1370,1350,1370,1390
1080 GOSUB 1450
1090 POKE v_edw,k:POKE v_zoku+1,1:MEM$(v_msbt,2)=MKI$(i0)
1100 MEM$(v_bf,2)=MKI$(buff):POKE v_frwf,i:CALL m_csrw
1110 POKE v_frwf,0:IF PEEK(v_stop) THEN "erre"
1120 CALL m_saved:IF PEEK(v_stop) THEN "erre"
1130 ds=USR3(process$(proces-1)):IF PEEK(v_stop) THEN "errb"
1140 k=PEEK(v_dn):IF k<4 THEN DEVICE STR$(k)+":"+RIGHT$(STR$(3-PEEK(v_mac)),1)
1150 proces=proces-1:CHAIN MEM$(v_p256+&H81,PEEK(v_p256+&H80))
1160 '----- ERROR
1170 '
1180 LABEL "erre":RESTORE "d1":GOTO 1200
1190 LABEL "errx":RESTORE "d2"
1200 READ m$:PRINT:CREV 1:PRINT m$:CREV 0:PRINT:POKE v_stop,0:GOTO 1130
1210 LABEL "errb":RESTORE "reb"
1220 READ m$:PRINT:CREV 1:PRINT m$:CREV 0:PRINT:ds=INKEY$(1)
1230 POKE v_stop,0:GOTO 1130
1240 '----- SUB
1250 '
1260 LABEL "open"
1270 ds=USR1(fe$(1)):IF PEEK(v_stop) THEN RETURN
1280 fe$(1)=RIGHT$(fe$(1),PEEK(v_yen)):IF fe$(1)="" OR fe$(1)="/" RETURN
1290 k=PEEK(v_mac):d=INSTR(fe$(1),".")
  
```



```

1300 IF (k=1 OR k=3) AND d=0 THEN fcs(1)=fcs(1)+".DIR"
1310 POKE v_sbdr,2:POKE v_op,3:d$=USR0(fcs(1))
1320 MEM$(v_fszl,4)=CHR$(0,0,0):MEM$(v_fnaml+46+22,5)=CHR$(0,0,0,0,0)
1330 RETURN
1340 '
1350 LABEL "ms"
1360 k=1:i0=1024:i1=&H10:d=0 :RETURN
1370 LABEL "x1"
1380 k=1:i0=256 :i1=&HC0:d=&HFF:RETURN
1390 LABEL "m2"
1400 k=2:i0=1024:i1=&H10:d=0 :RETURN
1410 '
1420 LABEL "poke"
1430 d$=USR2(MKI$(p)+MKI$(1)+CHR$(j)):p=p+1:RETURN
1440 '----- DATA
1450 LABEL "mem"
1460 MEM$(&HEE80,16)=HEXCHR$( "EB 5E 23 56 23 4E 23 46 23 CD 93 EE 13 0B 78 B1" )
1470 MEM$(&HEE90,16)=HEXCHR$( "20 F7 C9 7E C3 27 E0 00 00 00 00 00 00 00 00" )
1480 MEM$(&HEE95,2)=MKI$(m_lddea)
1490 d$=USR2(MKI$(buff)+MKI$(i0)+CHR$(d))
1500 i=PEEK(v_mac):IF i=1 OR i=3 RETURN
1510 '----- for MS-DOS
1520 RESTORE 1600
1530 p=buff :FOR i=0 TO 11:READ j:GOSUB "poke":NEXT
1540 p=buff+32:FOR i=0 TO 11:READ j:GOSUB "poke":NEXT
1550 p=buff+26:j=PEEK(v_crs):GOSUB "poke":j=PEEK(v_crs+1):GOSUB "poke"
1560 p=buff+58:i=PEEK(v_csdire)+PEEK(v_dn):IF i=0 THEN 1590
1570 h=v_csdire+26+8*PEEK(v_dn)+2*(i-1)
1580 j=PEEK(h):GOSUB "poke":j=PEEK(h+1):GOSUB "poke":RETURN
1590 j=0:GOSUB "poke":j=0:GOSUB "poke":RETURN
1600 '
1610 DATA 46,32,32,32,32,32,32,32,32,32,32,16
1620 DATA 46,46,32,32,32,32,32,32,32,32,32,16
1630 '----- MESSAGE
1640 '
1650 LABEL "d1":DATA エラーが発生しました!!
1660 'LABEL "d1":DATA Error !!
1670 LABEL "d2":DATA ファイル名を指定して実行してください
1680 'LABEL "d2":DATA What file-name?
1690 LABEL "reb":DATA リブートできません
1700 'LABEL "reb":DATA reboot error

```

リスト4 RD.X1

```

1000 'RD (RMDIR) ver 1.0 By M.Kameda
1010 '
1020 DEFUSR0=m_opens:DEFUSR1=m_preop:DEFUSR2=m_tranr
1030 '
1040 POKE v_dn,PEEK(s_dn):POKE v_mac,PEEK(s_mac4+PEEK(s_dn))
1050 IF fcs(1)=" THEN "errx"
1060 POKE v_od,1:GOSUB 1270:k=PEEK(v_stop):POKE v_stop,0
1070 IF k=0 THEN "erry" ELSE IF k>3 THEN "erre"
1080 GOSUB 1320:IF fcs(1)=" THEN "errx"
1090 k=PEEK(v_stop):IF k=3 THEN "errz" ELSE IF k THEN "erre"
1100 CALL m_dlfat:CALL m_dldir:CALL m_clos2
1110 '
1120 d$=USR2(process(process-1)):IF PEEK(v_stop) THEN "errb"
1130 k=PEEK(v_dn):IF k<4 THEN DEVICE STR$(k)+":"+RIGHT$(STR$(3-PEEK(v_mac)),1)
1140 process=process-1:CHAIN MEM$(v_p256+&H81,PEEK(v_p256+&H80))
1150 '----- ERROR
1160 '
1170 LABEL "erre":RESTORE "d1":GOTO 1210
1180 LABEL "errx":RESTORE "d2":GOTO 1210
1190 LABEL "erry":RESTORE "d3":GOTO 1210
1200 LABEL "errz":RESTORE "d4"
1210 READ m$:PRINT:CREV 1:PRINT m$:CREV 0:PRINT:POKE v_stop,0:GOTO 1120
1220 LABEL "errb":RESTORE "reb"
1230 READ m$:PRINT:CREV 1:PRINT m$:CREV 0:PRINT:d$=INKEY$(1)
1240 POKE v_stop,0:GOTO 1120
1250 '----- SUB
1260 '
1270 LABEL "open"
1280 d$=USR1(fcs(1)):IF PEEK(v_stop) THEN RETURN
1290 fcs=RIGHT$(fcs(1),PEEK(v_yen))
1300 POKE v_sbdr,0:POKE v_op,1:d$=USR0(fcs):RETURN
1310 '
1320 LABEL "dopen"
1330 i=INSTR(fcs(1),"/"):IF i THEN 1380
1340 d$=USR1(fcs(1)):IF PEEK(v_stop) THEN RETURN
1350 k=PEEK(v_dn):j=PEEK(v_csdire)+k:POKE v_fnaml+46+43,j
1360 i=v_csdire+26+k*8+(j-1)*2:POKE v_fnaml+46+44,PEEK(i),PEEK(i+1)
1370 POKE v_sbdr,2:POKE v_op,2:d$=USR0(fcs(1)):RETURN
1380 '
1390 fcs="":WHILE i
1400 fcs=fcs+LEFT$(fcs(1),i):fcs(1)=RIGHT$(fcs(1),LEN(fcs(1))-i)
1410 i=INSTR(fcs(1),"/")
1420 WEND:IF fcs(1)=" THEN RETURN
1430 d$=USR1(fcs):IF PEEK(v_stop) THEN RETURN
1440 GOTO 1370
1450 '----- MESSAGE
1460 '
1470 LABEL "d1":DATA エラーが発生しました!!
1480 'LABEL "d1":DATA Error !!
1490 LABEL "d2":DATA ファイル名を指定して実行してください
1500 'LABEL "d2":DATA What file-name?
1510 LABEL "d3":DATA ディレクトリにファイルがあります
1520 'LABEL "d3":DATA File exists
1530 LABEL "d4":DATA 指定されたディレクトリがありません
1540 'LABEL "d4":DATA No directory
1550 LABEL "reb":DATA リブートできません
1560 'LABEL "reb":DATA reboot error

```

す。エラーは1カ所にまとめて同一の処理がなされます。

MDではMD独自のマシン語プログラムを持っています。これはディレクトリ領域初期化の高速化のためです。そして、このように短いマシン語プログラムを使う場合は、EE00番地からの256バイトを使うことになります。ここは汎用ワークエリアなので保存はしておきませんが、一時的に置いておくことはできます(ほかの外部プログラムでもこうしていくつもありです)。

その後ろにはMS-DOS用の特別初期化ルーチンが続いています。実際の書き込みは「mcrsrw」ルーチンをコールすることで行われます。そして、エラーがなければ「msaved」ルーチンでいま書き込んだディレクトリをCLOSEします。これらのルーチンは、ただコールしただけじゃ正常には動きません。その前後で盛んにPOKEしているように、あらかじめ値を設定しておかなければならないのです。POKEアドレスの意味は7月号のアセンブルリストを見ればわかるでしょう。

そして最後はリブートルーチンです。USR3命令からCHAIN命令までがそうで、これはRDでも同じです。USR3で親プロセスを引数にして、その結果はVP256+&H81からに格納されています。この内容はフルパスファイルネームです。ディスクが入れ替えられている可能性も考慮して、DEVICE命令も実行しています。

RDでも基本的な作りは同じですが、OPENとCLOSEの部分が違っています。OPENが2つに分かれているのは、「ディレクトリ自体のOPEN」と「そのディレクトリ内にファイルがあるかどうかを調べるOPEN」があるからです。CLOSEの場合は、ファイルを消すのとわけが違って、3回に分けたコールが必要になります。そのほかには、初期化する必要もないのでMDのようなマシン語ルーチンはありません。

これで外部コマンドの概要はわかってもらえたと思います。まだ作るにはいたらないかもしれませんが、わかるころを改造してみるのもいいでしょう。来月はもっと突っ込んだ説明をして、なにか新しいコマンドを発表しながら、実際にコマンドが作れるようになるくらいまではやりたいと思っています。

対戦ポピュラス

祝一平VS西川善司

実況・解説 浦川博之



Illustration: Yamada

編集室で対戦ポピュラスなんかやられちゃ面白くって大迷惑。なのに西川君が祝氏に挑戦状をFAXで送っちゃうんだから、さあ大変。100号記念なのに、もっと実のある企画はないのかあ〜、といいつつOh!X史上最大の決戦の火蓋は切って落とされた。

5月上旬のある日。そもそも編集室にはX68000が2台並んでいるのが悪い。これで対戦ポピュラスをやるなっただって無理というもの。かくして今日もスタッフの対戦が行われるわけです。なかでもズバ抜けて強いのが西川善司。270面を制覇し、対戦は負けたことがないとか。

善「まあ、ぼくにかなう人はいないかな」
編「いや、祝さんがAmigaで始めて、いま420面だからわかりませんよ。ね、祝さん」
善「フフフ、負けませんよ、祝さん」
祝「(ニヤッと笑って中指を立てる)」

すでにこの会話以来、2人の対決は必然だったのです。

*

5/28 18:30決闘当日。

善「米た、祝さん」
祝「……いたな、青二才めが」
善「ひょっとしてあのFAX、怒ってる？」
祝「叩き潰してくれる」

おおっと、出会い頭にこのエキサイトぶり。おや、観客の中に丹明彦さんの姿が。丹さん、丹さんは善司くんを負かす寸前まで追いこんだそうですね。

丹「ええ、向こうが何もできなくなるところまでいったんですが、いつの間にか逆転されてしまいました。はは」

お、祝氏が自分のマウスと専用マット(なぜか航空機力学の本)を持って現れた。

善「道具まで気にしちゃって、もう」
といいつつ、善司くんもマット代わりのフロッピーケースを取りに戻っている。

さて、今回の対戦のマップはレビューを書いた中野修一氏が作った特製だということです。中野さん、ちょっとすいません、

どんなマップか教えていただけます？

中「ええ、いろいろあります」

へえ、たとえば？

中「(ニヤリと笑って) 結構スゴイです」

うーん、この人も意味不明な気合いが入ってるな。さて……。

中「どのマップにします？」

祝「じゃあ、この砂漠のにしよう。異存はないな？」

善「どのマップでも同じですよ。へへ」

おおおとギャラリーが沸く。

ここでちょっとマップの説明を。地形は完全に対称で、お互い人口は1人ずつでスタート。奇跡はすべて起こせます。ひとつ変わっているのは、沼が「底無し」に設定されている点。普通は1人沼に落ちるとそのマスは平地に戻るんだけど、このマップではいつまでたっても人が落ち続けるというわけ。沼を作られたら最優先で直さないでマズいわけです。

20:53 さあ、ゲームスタート。

祝「あれ、人はどこにいるんだ？」

中「(ニヤリと笑って) え、いるじゃないですか」

どーん。中央にそれらしく島を作っておきながら、人はマップの隅に、しかも岩に囲まれて細々とテントを立てていた。これじゃあ思うように家を増やせない。確かにスゴいマップだ。

善「あーっ、ちくしょう。でいでい」

ああ、むりやりテントを城にする気か。中野氏が「あっあっあっ」と心配そうにも嬉しそうな悲鳴をあげる。

家を作らせてもらえない悪魔の民。ふらふらとさまよっているうちに……。

YOU LOST

SOCRE 2570

あーっ、なんと開始後1分で西川善司の連勝記録ストップ! あまりの情けなさにギャラリーは聞いた口がふさがらない。

丹「体力もないのに砂漠を歩かせたりするから……」

祝「うっはっは、口ほどにもない」

善「しまった。気にしすぎたあ」

中「だからあ、もうちょっと気合い入れません？」

狙いどおりの展開に嬉しそうな中野氏。気合いを入れ直して、20:55再開。今度は2人とも慎重に人が増えるのを待っているようです。

祝「死ぬなよ死ぬなよ……よおーし!」

城は小さい家に比べて人の増えるペースが速いが、収容人員が多いため人があふれるのには時間がかかる。ということは、「城をとるとき壊して人を追い出す」というのが常用テクニックになるわけです。2人は次々と城を作っては追い出し平地を開拓しています。左上のマップを見ていると、平たい大陸がじわじわ中央に向かって伸びていくのがズキミ。

スゴゴゴゴ。おおーっと、祝氏の領土に地震。最初にしかけたのは善司くんのだ。

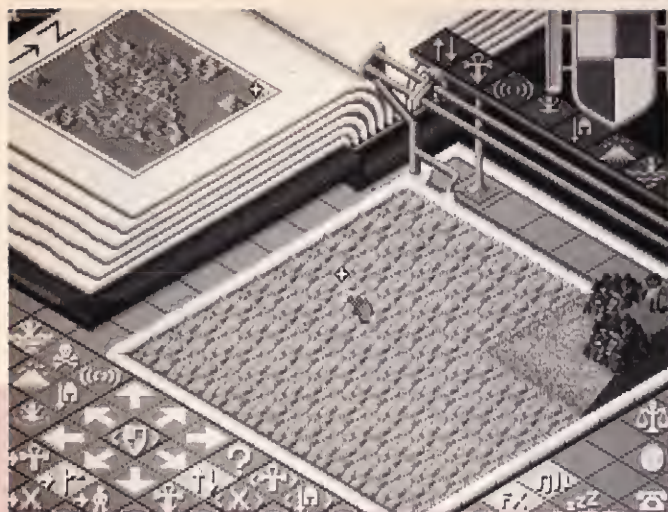
祝「地震なんか効かないもん」

さっさと修復してしまう祝氏。マウスのクリックにムダがない。さすが420面はダメじゃないぞ。

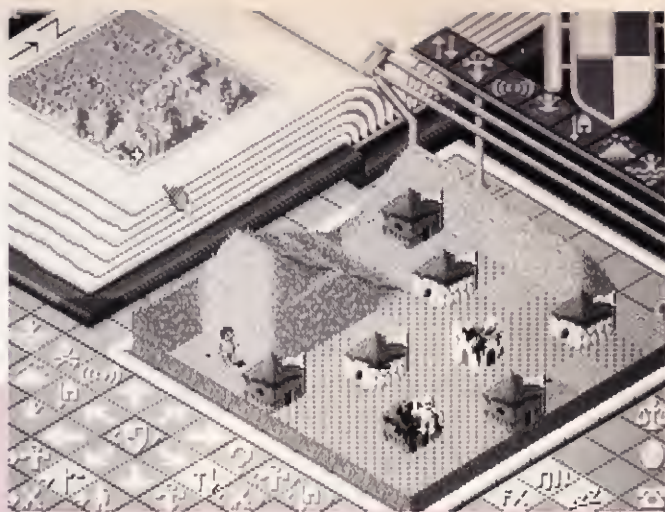
Oh!X通巻100号に寄せて

ども。ポピュラスやりたさにパソコンを買ってしまった浦川です。いま100面ちょっとですけどね。え? 機種ですか? よ、弱ったな、なんでもいいじゃないですか。

まあ、自宅でじっくりとコンピュータをイジめるのもいいですが、人間相手となるとまた格別。うひ。Oh!X編集部にはたくさんX68000がありまして(当たり前だ)、その中には2台RS-232Cでつながれたヤツもあります。これを見ると、さもX68000が「ぼくたち対戦ポピュラスのためにここにいます」と言っているような気がして、つついそこら辺の人に「やろーやろー」と声をかけてしまう。かくして編集者がドアを開けて入ってくるなり、締め切り間際のライターがマウスをカチカチやっている姿に頭をかかえるという日々が続くのでした。おっと、これのどこが100号記念の祝辞なんだろう。というわけで合掌。



これが1回戦のマップの初期状態。左右対称の大陸と丹念に岩が配置されている。ところどころに沼も置いてあるという。ちなみにリーダーはいない。



戦況はかなり煮詰ってきた。かつての海はどこへやら……。下が祝一平、上が西川善司。手作りの山攻撃が城や町を破壊している。もう泥沼。

ゴボシ あれ？
ゴボシ 何の音だ？
ゴボシ ぬ、沼だ！

だはははと無責任に笑うギャラリー（なぜか驚嘆より笑いが先に立ってしまう）。

祝「やりおったな」

善「いえ、なんにもしてませんよお」

祝「と、いうことは……」

ギャラリー全員の視線が中野氏に集まる。

善「まさか、最初っからあるんじゃないか」

中「まあ見てのお楽しみ」

やはり中央の島に沼があった。新大陸に家を建てようと勇んで出かけた民は、この沼に沈んでいたわけ。中野氏恐るべし。

祝「うーん、砂漠の沼は発見しづらい」

善「中野さん、あらかじめ設定しとくなんてすごいイジワル（スゴゴゴゴ）」

しっかりスキを見て地震を起こす善司くん。

丹「砂漠で地震は効きますよ」

へえ、なんで？

丹「ほら、外をうろちょろしてる間に体力がなくなっちゃうから」

善司くんはマナが貯まるたびに地震をしかけます。対照的に祝氏はマナを貯めながらひたすら領土を拡大。

21:35 ふたりの領土がそろそろ接してきました。善司くんが一番敵地に近い家を探して、画面の端にくるように設定している。

カチカチカチカチ……

みるみる相手の土地が盛り上がる。ワッハハハと無責任に笑うギャラリー。でた。これが善司くんの得意技、手作りの山だ。

祝氏は地震で素早く取り壊す。しかし修復し終えたところにはマップのほかの場所ですんずんと巨大なピラミッドが立っている。祝「むう（シユイイン）」

おーっと、怒った祝氏が火山をお見舞いだー！ しかも二段重ね！

*

22:00 あれから1時間。手作りの山と火山が乱れ飛んで、かつての平地はどこへやら。家の数を見ると善司くんのほうが押し気味ではあるけど、人口ゲージを見るとまだまだ互角。人数が多いので次第に処理速度も落ちてきた。しかもハングアップ防止のため2400ボーでやっているのではなさ。マウスの反応が悪くてときどきヘンなところがぼこっと盛り上がりつつあります。

22:30 開始から1時間半たって戦いはやや膠着状態に。そろそろ休憩にしません？ 祝「向こうが泣いて頼むんだったら休んでやってもいいよ」

善「もう、祝さんったら強情なんだから。素直に休みたいと言えればいいのに」

祝「なに、そんなに休みたいの？」

善「まさか。祝さんが泣いて頼むんだたらべつですけど」

次第に善司くんがじわじわと平地を獲得している。やはり手作りの山の対応に追われ続けている祝氏の不利は否めない。ところで祝さんが手作りの山はほとんどしかけないのは、なにに信条があるのだろうか？

祝「おい、休んでやってもいいよ」

善「いいですよ、べつに」

祝「……休んでやってもいいんだよ」

善「だからいいってば（ブォン）」

あーっと、騎士が誕生。対戦ではよほど有利でないとできない行為だ。散在する祝氏の家を焼いてまわる騎士。さらに手作りの山攻撃が襲いかかる。これらを全部修復しながら挽回をはかるのは祝氏といえども至難の技だ。

祝「むっ。くそっ。くそっ」

脂汗をにじませながら力をこめてクリックを続ける祝氏。反応が鈍いんだから、そんなに力をこめたって……。

祝「うるさい。やってるほうの身にもなってみろ」

ついにいっぱいだった人口ゲージも減少を始めた。騎士が次から次へと送りこまれ、あっちこっちで山が立つ。祝氏側の家は端のほうに散在するばかり。

そして23:07。

祝「……うむ。今日のところは負けにしていあげよう」

ついに祝氏敗北宣言！ 西川善司のTKO勝ちで決着！

祝さん、敗因は？

祝「若さに負けた」

2時間20分の長丁場ですからね。

祝「それから、あの沼は発見しづらいからキライ。そもそもマップを作ったあのコミッショナーが悪い」

勝った善司くんは？

善「そうねえ、へへへ。まあ、丹さんのほうが強かったかな。なんちて。ぼっくん」

祝「この借りは必ず返すぞ」

善「いつでも来なさい。はっはっは」

*

その4日後。

「ちわーす」編集室に入っていくと、さっそく再戦している2人の姿があった。

善「祝さんが泣いて頼むからさあ」

祝「この前のは練習。今度が本番」

2人とも好きにしよう。い

今度はもっと素直なマップで対戦。雪原に点対称に日本が2つ配置され、沖縄に1人だけ人間がいるという設定です。

おや、祝氏が家をくずして、一番低い平地で展開するのに対して、善司くんは一段

高いところで展開している。洪水対策か？丹（また見に来ている）「いや、やりこんだ人なら洪水は使いません。火山を何発も起こしたほうが有効ですから」

高い土地をいじるほうがマナがいるんですよ。マナの少ない序盤にこういうことをしていのかなあ。

19:45 やはり人口比7:3ぐらいに差がついて、今度は祝氏が中央部を押さえた。苦しい善司くん手作りの山で反撃！ また泥沼の戦いが始まる。立てる崩す、立てる崩す、立てる崩す、沼にはまる。

善「やっぱり沼が奇跡のなかでは一番有効ですからね」

祝「えっ？ 沼の弱点知らないの？」

善「……そんなこと言って動揺を誘おうとしてるんでしょ」

祝「そう思う？」

直接対戦ならではの口頭の戦い。

20:17 祝氏がメガネをはずした。気合いの入れ直しか（どうでもいいが、氏はサングラスがとても似合うお方である）？

お互いの境界にまんべんなく山が立っている。やはり山の被害のせいか、祝氏のリーダー幅が縮んだような。

「シユイイイン」あ、火山だ、祝氏が火山をおみまい！ さらにシンボルを移動にかかる。ここで一気に攻勢に出るのか。

祝「あれ、できない」

リーダーは敵陣との境で死んでいた（笑）。

善司くんは山を作って、相手の復旧の間に領地を広げる作戦に、祝氏はリーダーを誘導して個別撃破の作戦に出ています。

20:33 祝氏のリーダーは合体を繰り返し、パワーのある奴になりました。楽しげに誘導先を選ぶ祝氏ですがその途端……。

ゴボシ

祝「……！」

リーダーのいたところには沼が広がっていた。ギャラリーが無責任に笑う。

祝「……（シユイイイン）」

善司くんの領土に怒りの火山が炸裂！

21:10 そろそろ勢力が五分五分というところ。やはり善司くんは攻勢にたけています。おっと、何を考えたか善司くんが自分の領土に地震をしかけました。

善「こうやってシンボルに人を集めるんですよ」

恐るべき早さで最強の騎士が誕生。さらに騎士が敵地に向かっている間にも手加減しない善司くん。

善「ああ、祝さんたら僕に無断でこんなところに城を（カチカチカチカチ）」

山を立てている間に騎士が祝氏の領土に到着！ が、祝氏は慌てずに騎士の周りに穴を掘り、騎士を水の中に沈めてしまったあ。もがく騎士。体力が少しずつ落ち始める。善司くんはぜんぜん気がついていない。ギャラリーは笑いたいのを必死にこらえています。そのまま何事もなかったかのように自分の領土を整備している祝氏。数分してふと善司くんが右上のウィンドウを見ると……。

善「ああっ、なんかもがいてるう」

だはははと爆笑するギャラリー。たちまち敵住民を池に落とすという「水攻め攻撃」が乱れ飛びました。

22:27 山を残しながら、自分の領地はしっかりキープしている2人。しかしやはり中心部は善司くんが取り、祝氏は周辺部に追われています。自分の領地に地震をしかけている善司くん。出てきた人間を、シンボルのある敵陣まっただなかに集合させる

「一方的ハルマゲドン」攻撃です。騎士同様の追い込み技ですね。

しかしそれでも事態は終結しない。千日戦争状態にあると判断した中野氏が、善司くんにハルマゲドンを起こすよう指導勧告。以後善司くんは奇跡を起こすのを控え目にして、マナの集積をはかる。一方祝氏は、再びマウスを汗だくでクリック。

祝「もーいや、こんな生活」

挽回はできなかったが、この抵抗が効いて善司くんがハルマゲドンを起こすまでにはさらに1時間を要したのだった。

23:45 「ウホウホウホ」ハルマゲドンスタート。人口ゲージは祝氏の圧倒的不利を伝えている。ああ、やはり祝氏も善司くんの独走を止められなかったか。画面の中で2人のリーダーが向き合った瞬間！

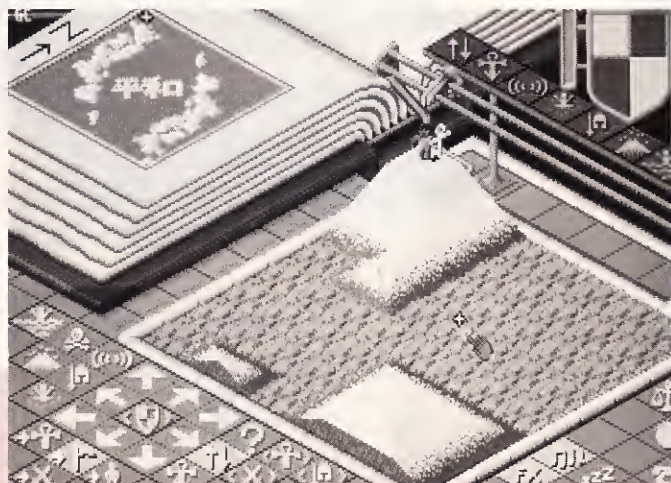
ピタ！

うおお、ハングだあ！ 天は祝氏に武士の情けをかけようというのかー！

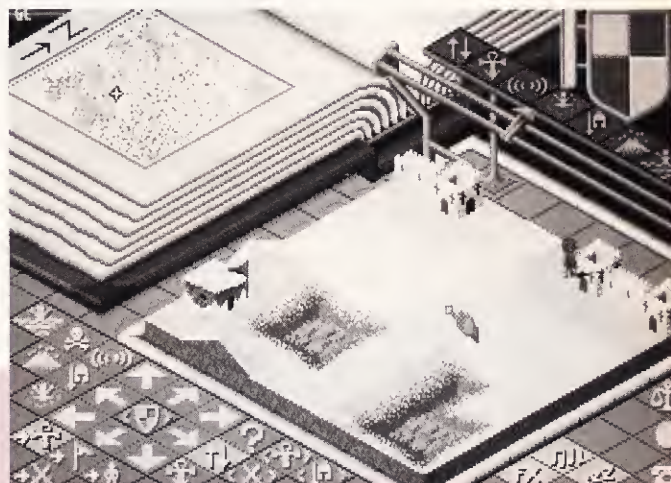
結局波乱のラストを乗り越えて、西川善司のハルマゲドン勝ちが決定しました。

結局善司くんの2勝という形になりましたが、聞いたところでは祝氏是对戦がこれで3回目ということですから、いかに420面まで進んでいても、対戦ポピュラスのノウハウのある西川善司くんに一日の長があったといえるでしょう。

しかし、この対戦もさらなる戦慄の歴史の序章に過ぎないのです。このあともさらに西川善司対中野修一などの数々の恐ろしい戦いが、編集室では繰り広げられています。対戦ポピュラスは確かに面白い。時間は使うし電話代もかかるし友人関係も下手するとこわれる。それでも対戦ポピュラスは面白い。あなたはこの面白さにつかってみる勇気がありますか？



2回戦。中野氏による平和島マップ。祝氏のリクエストで気候は水河時代となった。今度はなんの仕掛けもない。赤い敵が樺太から……(ちょっとあぶない)。



下が西川氏が上が祝氏。画面上のあちこちにポツポツと穴が見える。ちまたでは「温泉」と呼ばれている。善司くんの地震突撃攻撃対祝氏の執拗な沼攻撃。

X68000 10万台突破記念

愛読者特大 モニタプレゼント

Oh!Xは通巻100号なんだよ〜、とはしゃいでいたら、ほとんど時期を同じくしてわれらがX68000が10万台出荷を達成した。これぞ歓喜の2段重ね！ここはひとつシャープさんをお願い！というわけで豪華プレゼントを提供していただきました。どうです、スゴイでしょ。特に大型ディスプレイやカラーイメージスキャナなんて持っている人、少ないんじゃないかな。えっ、本体はないのかって？だって大部分の皆さんはすでにX68000ユーザーじゃないですか。それに周辺機器ならX1/turboユーザーでも使えるでしょ。なに、X68000に乗り換えたい？だったら本体ぐらい自分で買わなきゃね（というのがOh!Xの本音なのだ）。なお、9番以外はモニタプレゼントだから、当たった人には感想文をお願いします。

2 熱転写カラー漢字プリンタ

CZ-8PC4

99,800円

1名

48ドット、7色のカラー印字ができるプリンタ。もちろんグラフィックもプリントできるぞ。いろいろなカラーリボンも使える。



21型カラーディスプレイ

CU-21HD 148,000円 1名

着脱可能なスピーカーを搭載した大きなカラーディスプレイ。これでゲームをやったらさぞかし気持ちいいことでしょう。



3

カラーイメージ スキャナ



CZ-8NS1 188,000円 1名

最大A4サイズの絵や写真をフルカラーで読み込むことができるカラーイメージスキャナだ。

4

サイバースティック

CZ-8NJ2

23,800円

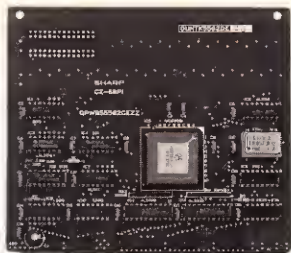
1名



ゲーム命の人ならば、ぜひ手にいれてほしいアナログジョイスティック。細かな操作も行いやすくなるぞ。

5

数値演算 プロセッサ ボード



CZ-6BP1 79,800円 1名

面倒な計算やレイトレーシング、シェーディングなどの処理速度を一気に高めることができるこのボード、CGには最適。

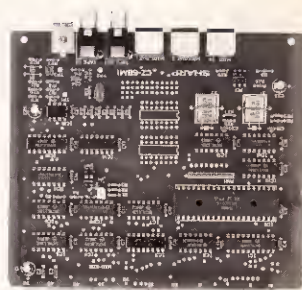
6

MIDIボード

CZ-6BM1

26,800円

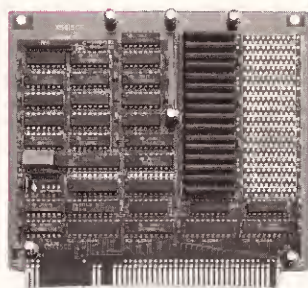
1名



最近はいろいろなゲームもMIDI対応になっている。このボードがあればMIDI楽器が接続でき、鮮やかなサウンドが楽しめる。

7

2MB増設RAMボード



CZ-6BE2 79,800円 1名

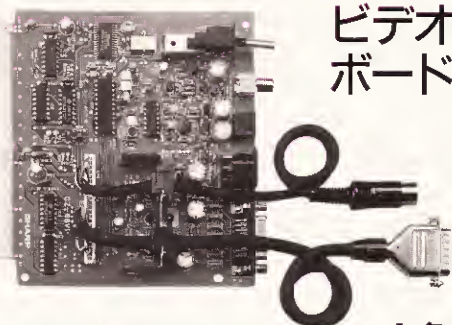
あれもこれもパソコンでやりたい、という人はRAMボードの増設は必至。そんなあなたにこのボードをプレゼント。
(1M増設済のこと)

プレゼントの応募方法

とじ込みのアンケートはがき(ただし、今月のもの)の該当項目をすべてご記入のうえ、希望するプレゼント番号をはがき右下のスペースにひとつ記入してお申し込みください。締め切りは1990年8月18日の到着分までとします。当選者の発表は1990年10月号で行います。

8

ビデオ ボード



CZ-6BV1 21,000円

1名

このボードを使えば、X68000で作ったグラフィックや、プレイしているゲームなどが、簡単にビデオに録画できるようになるぞ。

(以上、シャープ提供)

9

キーボード延長ケーブル

1,980円

黒/グレイ
各5名



九十九電機より創刊100号を記念して、オリジナルのキーボード延長コードをプレゼント。寝っ転がってキーボードも打てるかな。

6月号プレゼント当選者

1A)ジェミニウイング(沖縄県) 宇良秀樹 B)闇の血族(静岡県) 野村一洋 2ねじ式(千葉県) 田浦達也(山口県) 大隅研治 3The File Professor(東京都) 高橋信博(静岡県) 戸塚昭信 4サイクロンEXPRESSα(秋田県) 佐々木仁志(神奈川県) 鈴木利明(東京都) 三田恭一郎(静岡県) 三橋和美(大阪府) 藤沢直樹 5A)FAR SIDE MOON(広島県) 本谷正樹(愛媛県) 横山智生 B)A列車で行こうII(富山県) 加賀見政和(長崎県) 佐藤充浩 C)大海令(埼玉県) 桑原智志(岡山県) 梅田敬 D)南海の死闘(東京都) 小山薫(広島県) 岸本秀生 6コース(北海道) 加納一郎(福島県) 村上健(京都府) 村久木康夫 7ジャック・ニクラウス・テレフォンカード(宮城県) 伊藤洋美(東京都) 平尾雄一(神奈川県) 長嶺隆(奈良県) 野瀬正博 林衛 8スタークルーザー X68000用(福島県) 岩淵正樹(東京都) 大橋飛雄吾 Xlturbo用(神奈川県) 田口聡(岡山県) 小谷恒 9キューブランナー(東京都) 角野俊人(神奈川県) 武藤俊哉(京都府) 田中啓 10レナム(岩手県) 片岸健一(群馬県) 石山篤志(兵庫県) 郡茂樹(新潟県) 霧島博史(香川県) 佐竹勝博 11A)ガンマ・プラネット(東京都) 高橋明(群馬県) 藤田明(愛知県) 永井周作 B)グランディフロラム(千葉県) 久原義弘(栃木県) 佐藤崇(三重県) 大橋隆太郎 C)Simple-CAD X68K(福島県) 仲山秀樹(和歌山県) 辻本浩一 12上海II(長野県) 吉沢克明(兵庫県) 堀江良孝 13ボビュラス(千葉県) 佐藤一成(島根県) 原誠(鹿児島県) 園田光太郎 14プログラムオペレーティングシステム(東京都) 木部幸雄(石川県) 川口聡 15PIO-6BE1-A(東京都) 飯塚晃太郎 16銀河英雄伝説+set(埼玉県) 武藤一文 加藤勲(京都府) 牧本隆 17G68K II(東京都) 信川洋(福岡県) 平山謙司(宮崎県) 土井順之 18A)D-RETURN(神奈川県) 細井実人(茨城県) 伊東臣明 B)ずるかまし(宮城県) 坂井一弘(東京都) 千葉広道 19A)オリジナルコーヒーカップ(北海道) 飯田伸一(愛知県) 五月女優(広島県) 田村和廣 B)ツインビー(茨城県) 内田好則(京都府) 上野政幸 20バトルチェス(三重県) 水谷泰三 21A)Zero(愛媛県) 武智和彦(鹿児島県) 本真光 B)Misty3(茨城県) 地引秀和 原田大輔 22セレクトッドソーサリアン 1(長野県) 塚本隆司(岡山県) 横山博道(福岡県) 浜地啓 2(東京都) 松村一朗(神奈川県) 三沢弘之(山梨県) 深沢享広 3(茨城県) 程田勝也(兵庫県) 村上貴之(大阪府) 中山良樹 23ウインドブレイカー(北海道) 渋谷康則(東京都) 八木貴弘(神奈川県) 久崎圭(岐阜県) 山口忠(大阪府) 鈴木哲也 24「この木なんの木」のCD(茨城県) 染谷祐一(福岡県) 徳久雅人(大分県) 山田博

以上の方々が当選されました。おめでとうございます。商品は順次発送いたしますが、入荷状況などにより遅れる場合もあります。また、公正取引委員会の告示により、このプレゼントに当選された方は、この号の他の懸賞には当選できない場合がありますのでご了承ください。

(価格はすべて消費税別です)

ポケコンでCARPGを

Matsui Shin
松井 信

おっと、100号記念にちなんでポケコンの記事も復活かな？
でも何をやるかという、実はテーブルトークのRPGを楽しむのに利用しちゃおうというお話なんです。使用するのは圧倒的シェアを誇るPC-E500シリーズです。お楽しみに。



CARPGとは、Computer Aided Role-playing-game、つまり、コンピュータを利用したRPGのことです。私がいま名づけました。コンピュータRPG（以下CRPG）ではありません。あくまでもテーブルトークRPG（以下テーブルトーク）のサポートを目的としています。

テーブルトークRPGとはなにか？

テーブルトークとは、机の上で多人数でやるコンピュータを使わないRPGです。というよりは、CRPGのほうをコンピュータ上でやるテーブルトークの真似ごとといったほうが正確です。

D&D(Dungeons&Dragons)などのテーブルトークは、最近になってようやくやっている人も増えてきたようですが、それでも実際にやったことのある人はまだ少ないようで、RPGといえばCRPGのようなゲームと思っている人も多いようです。しかし、CRPGはテーブルトークから戦闘システム部分とストーリー進行を抜きだしたもので、それはテーブルトークの楽しみのごく一部に過ぎません。

テーブルトークの楽しさとは基本的にロールプレイ、すなわち「ゴッコ遊び」の楽しさです。つまりRPGというからには、キャラクターを演じられることが必要です。

CRPGでは、キャラクターを動かしてこそすれ、演じているとはとうていいえません。ドラクエをしていて自分が(本当に)勇者だと思いながらやっている人はたぶんいないでしょう。

しかし、テーブルトークでは、あなたはガラスの仮面のごとく、完全にキャラクターになりきって、現実世界のようにファンタジーワールドの中を冒険することができるようになります。いくつかの作業と若干の想像力を必要としますが、こういったリアリティと面白さはCRPGの比ではありません。

テーブルトークの実際

とはいえ、テーブルトークにも問題点があります。ひとつは、1人ではできないという点、しかも、そのうちの1人は「マスター」と呼ばれる進行役にならなければいけません。そして、ある程度の時間(数時間以上)と、場所(人数+機のスペース)が必要です。そういえば、マニュアルとそのほか道具も必要です。

テーブルトークはCRPGのように買ってきてすぐにできるものではありません。

とにかく、マスターになる人が、シナリオと呼ばれる台本(のようなもの。ゲームの設定およびストーリーなどを書いたもの)によって、ゲームを進行し、その架空世界のすべての出来事を管理し、同時にプレイヤーの不条理な要求に対処するわけです。当然、かなりの負担がかかるので経験者が望ましいわけです。

一方、1人ひとりのプレイヤーは、「キャラクター」というゲーム上での仮人格、つまり、その世界での自分を持ちます。それには、強さ、魔法、持ち物、その他さまざまな属性が決められていて、その世界におけるキャラクターの個性を表し、その行動に一定の制限を与えます。この辺はCRPGと一緒にですが、CRPGでは戦闘に関係ない属性はほとんどないのに対して、テーブルトークには戦闘以外にもさまざまな属性が存在します。キャラクターというのはひとつの人格なのだから、これは当然でしょう。

以上、テーブルトークのいい点として、

- 1) 別人格を演じることができる
- 2) 実際にはない世界で遊ぶことができる
- 3) 破壊衝動(?)を満足でき、ミッションに成功したときはカタルシスが得られるということがあげられます。また、
- 4) 議論や会話の訓練になる
- 5) 多人数でわいわい遊べる
- 6) マスターになって、いいシナリオがで

きたときは自己顕示欲(?)を満足できるなどのメリットも忘れることができません。

これだけの利点を持つテーブルトークが、ボードゲーム界に与えた影響は大きく、SLGなどは駆逐されかかって、SLGの雑誌であったタクティクスなどは、本家が季刊になって、月刊のRPG雑誌を出しているほどです。

CARPGとは

前に述べたように、やはりマスターは大変です(同時にやりがいもあるが)。そうしたある日、疲れたマスターである私は、ひたすら作業をしていた思い出。

テーブルトークの問題点である「作業」は、多くは数値の処理という機械的な作業です。これをコンピュータ化してしまえば、マスターの負担は軽減し、本来のロールプレイに専念できるようになるんじゃないか。これが、CARPGなのです。

テーブルトークにおける作業は、次のように分類されます。

- 1) キャラクターを作る
- 2) シナリオを作る
- 3) ゲームをする

まず、1)ですが、この辺は作業というよりは楽しみに属するものなので、ワープロの利用ぐらいにとどめておきます。

次に2)ですが、シナリオを作るというのは、小説のあらすじを作るようなものです。

まあ、仲間内でやるんだったらストーリーはどこからバクってくればいいのですが、敵の設定、地図作成、ストーリーの記述といったところだけでもかなりの作業となります。

これは、市販のシナリオを買ってくれば済む問題ですが、何千円もする高いものだし、そんなにたくさん出ていません。それに、自作シナリオを成功させることこそがマスターの醍醐味だし。というわけで、この辺のCARPG化はそのうち取り上げた

いと思います。

そして、なんといってもマスターがいちばん大変なのは、3)の実際のゲーム中でしよう（と私は思う）。

なにしろプレイヤーは何人もいるのにマスターは1人なのだから。戦闘場面でたくさんの敵キャラクタを操りながら、プレイヤーの受け答えをするのは、やっぱり大変なことです。たとえば、

プレイヤーA：ゴブリン6に3ダメージ！
マスター：はい。

プレイヤーB：オーガ3に12ダメージ！
マスター：はいよ。

プレイヤーC：魔法かけるよ。ホールドパーソン！ゴブリン4と5！

マスター：はい（コロ、サイコロを振る）。5は止まった。それから？

プレイヤーA：そっちの番だよ。

マスター：そうか。じゃいくよ。ゴブリン1が、えーと誰の前？あ、そう。アーマークラスいくつ？（コロ）当たった。えっとダメージは（コロ）2ね。じゃ、ゴブリン2は……。

これをえんえんと繰り返すのだから、慣れれば機械的にできるとはいえやっぱり面倒くさい。ましてや徹夜でやっていたりすると、うっかりするとパニックになりかねません。

そこで、戦闘中の敵モンスターのヒットポイントや攻撃を、コンピュータに管理させようというわけです。このプログラムを次回掲載する予定です。

コンピュータはなにを使う？

ところで、CARPGに使うコンピュータはなにがいいか。それは実はポケットコンピュータなのです。

まず、学校なんかでやるときは持ち運びができなくてはいけません。その点、ポケコンなら持ち運びもできるし、値段も安く、また、高級電卓として使えるので無駄な投資にはなりません。それに、工学系の大学生のほとんどはポケコンを持っているでしょう。

こういふと、ポケコンなんて、という人もいるかもしれませんが、今のポケコンをなめてはいけません。シャープのPC-E500（または、PC-1480U、PC-1490U）は、X1のBASICのような（というよりもN88-BASICのような）強力なBASIC、パソコンにも引けをとらない高速性、40×4行の広い画面、32KバイトのRAMは一部をRAMディスクとして使用でき、RS-232C

ケーブルでパソコンにつなげる、などとてもなく強力なマシンなのです。

では、自宅でやるならパソコンでいいやという意見もあるでしょうが、テーブルトークではマスターの情報はプレイヤーに見せてはいけないことになっています。したがって、机の上にマスターに向けてディスプレイが載ることになり、普通の家ではちょっと苦しいでしょう。そのため、ポケコンのほうが都合いいのです。

テーブルトークを始めるには

現在、たくさんの種類のテーブルトークが市販されていますが、やはりおすすめはD&Dおよび、AD&D(Advanced D&D)です。したがってこの連載も、対象は基本的にD&D、AD&Dとします。

D&Dはやはり日本ではもっともメジャーで、サプリメント（追加シナリオ、その他ゲーム補助用のツール）が多く、またルールがシンプルなため初心者でもやりやすいという特徴があります。

しかし、実は米英ではAD&Dのほうが遙かにメジャーで、そのサプリメントの量はD&Dの比ではありません。ルールもD&Dより体系化され、より面白くなっています。なにぶん英語というハンデがありますが、高校生でも読める程度のものですらそれほど心配することはありません。日本語版も7月から出版されるはずですが、最初は誤植が多いと予想されるので、いっそのこと英語版を買っても無駄にはならないでしょう。

というわけで、ようやくテーブルトークを始めるわけですが、なにもしたことがない人がいきなりマスターを始めるのは大変です。しかし、誰かがマスターをやらなけ

ればいけません。しかし、なにかとんでもない間違いをする可能性もあります。

そのため、まず最初は（少なくともマスターをやる人は）どこかでテーブルトークを体験してくることをおすすめします。たとえば、どこでも高校、大学なら誰かしらはテーブルトークをしているものですから、友達のつてから仲間に入ってみるというのがひとつの手です。

ほかに、テーブルトーク関係の雑誌には地域的なテーブルトークサークルのメンバー募集が出てますからそこに連絡を取ってみるという手もあります。

マスターへの道

とにかく、マスターになる人はロールプレイとはなにかということを理解しないといけません。ルールを読むことも忘れずに。

それから、特にファンタジー系テーブルトークの場合、たくさんのファンタジー小説と、ヨーロッパの歴史書、そしてそれ以外にもたくさん小説も読んで素養をつけておきましょう。マスターというのは、作家にして脚本家、監督にして俳優というとてもやりがいのある総合プロデューサーなのです。

それでは来月はプログラムに入ります。

＜参考文献＞

D & Dがよくわかる本 富士見文庫 490円

D&Dが具体的にどのような手順で進められるのかわかる。D&D初心者にはおすすめ。

眠れる龍 現代教養文庫 720円

アメリカのゲーマーの生活がわかってなんとなくほのぼのする。ファンタジー小説としてもいい出来。

ドリームパーク 創元推理文庫(SF) 580円

マスターの内輪うけと評されるだけあり、マスターをやっている人には面白い。

ゲーム紹介(1)

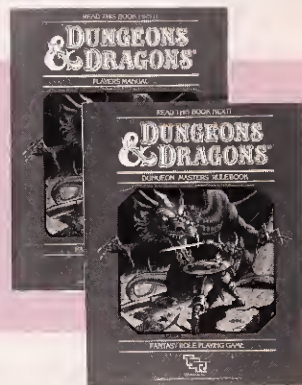
Dungeons & Dragons

RTS, Inc. (日本語版：新和)

テーブルトークといえばD&Dというぐらいメジャーなゲームで、特に日本ではほぼ主流となっている。とにかくルールが簡単で覚えやすく、初心者でもとりあえず20面サイコロを振って殴っているだけで十分楽しい。

しかしながら、古いゲームであるということ是否定できず、攻撃は最大の防御でありキャラクタのレベルが2桁になるころからなにか間違ったゲームへと発散していく傾向が多々ある。最高レベルである36のあとには、みんなで神様をやろうというルールまであるが、きつただの冗談だろう。

通称、赤Dといわれるベーシックと、青Dといわれるエキスパートの2つの箱が最低限必要。



上級セットとしてコンパニオン、マスター、インモータルの拡張ルールセットがある。

基本インタフェイス回路 その2

Misawa Kazuhiko
三沢 和彦

今回は製作実習編です。とても簡単な回路ですし、実体配線図も用意しました。注意事項も徹底的に詳しく解説してありますから皆さんも部品を揃えて実際に挑戦してください。うまくいったときの喜びは格別ですよ。

いよいよお待ちかねの製作実習編です。今月が待ち切れなくて、もう部品を揃えてしまった人もいられるかもしれませんね。とにかく、まずは部品表のとおり部品を揃えてください。



汎用ケーブルの製作

最初にジョイスティックポートと自作回路とをつなぐためのケーブルを作ります。このケーブルは1本作れば、連載で製作する回路すべてに使えるようにしてあります。圧着用の10ピンフラットケーブルとコネクタとは買ったお店で圧着してもらっておきます。部品を買うときに頼めば、その場で

圧着してくれるはずです。

圧着されたコネクタを見ると、一番端に印がついているでしょう。これが1番ピンです。さて、このフラットケーブルを9ピンDサブコネクタにハンダ付けしていきます。9ピンDサブコネクタはメスコネクタでなければ、X68000につなげないので注意してください。

そして、Dサブコネクタの表に出る側をよく見ると、小さく1~9の数字が記されているのがわかるでしょう。そこで、圧着コネクタの1番ピンにつながっている線から順番にDサブコネクタの各端子にハンダ付けしていくのです。Dサブコネクタの端子どうしの間隔が意外と狭いので、ハンダ

が隣に一つこないように注意してください。

ところで、ケーブルは10ピンでDサブコネクタは9ピンですから1本余ることになります。1番の線は9番ピンのGNDにいちじくにつないでおきます。ハンダ付けが完了したら10本のケーブルを束ねてDサブコネクタケースについている金具で止めて、コネクタ全体をケースに納めます。これで完成です。



基本 | O基板の製作

基板と回路を組むときにもっとも頭を悩ませるのは、部品の配置です。部品の配置をうまく決めるかどうかで配線の手間がまったく違います。皆さんは図1の実体配線図を参考にしながら、以下の説明を読んでください。

サンハヤトのICB-87という基板はIC1個用の汎用基板で、ICの足まわりの配線がしやすいように工夫されているものです。次回に製作するA/Dコンバータもこの基板上に作るのです。同様にまとめて買っておくのもよいでしょう。

●主な部品の取り付け

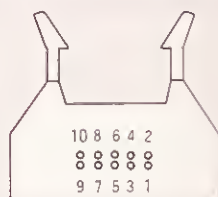
まず最初に、先ほど作った汎用ケーブルをつなぐ基板用コネクタを取り付けます。まずは10ピン全部をハンダ付けしてしまいます。このとき、ハンダ付け面から見て、ジョイスティックコネクタのピン番号は図2のように対応しています。ハンダ付けしたピンから各端子への配線はまだ行いま

部品表

9ピンDサブメスコネクタ
Dサブコネクタケース (DE-C1-J6)
10ピンフラットケーブル
10ピンコネクタ (PS-SRN10)
IC用基板 (サンハヤトICB-87)
10ピン基板用コネクタ (HIF3BA10P-DS)
16進ロータリースイッチ (アルプスSR2P)
ICソケット16ピン
74LS247
TLR313
抵抗510Ω
ビニール配線材

価 200円
価 360円
価 30円
価 300円
価 30円
価 100円
価 250円
価 25円
価 100円
価 210円
価 10円
計 1000円

図2 基板用コネクタ(ハンダ付け面から見た図)



ジョイスティックポート
1 IC402
2 IC402
3 IC402
4 IC402
5 IC402
6 IC402
7 IC402
8 IC402
9 GND
10 GND

図3 IC、ソケットを上から見た図

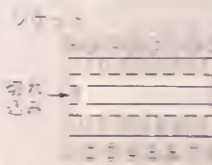
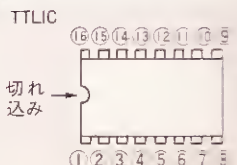
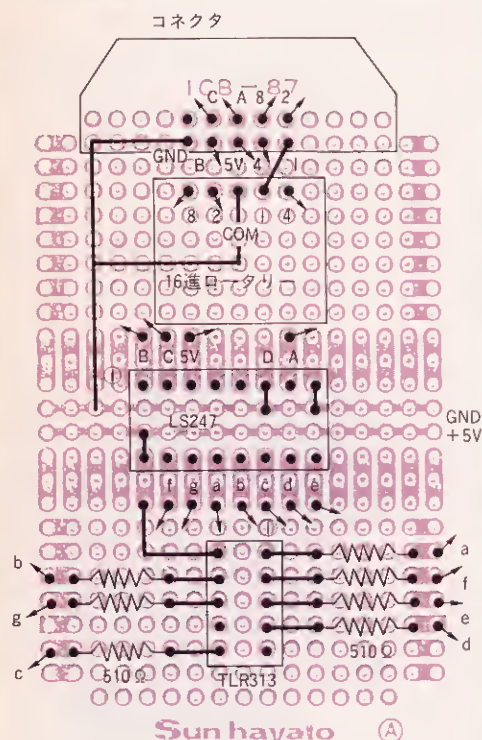


図1 実体配線図



↑ビニール線と同じ記号 (a-f, A-C, 1~8, 5Vライン) をつなぐ (ジャンパ線)
●基板上をメッキ線をつなぎ、ハンダ付け

せん。

次に、ICソケットを取り付けますが、ソケットを上からよく見ると図3のように片方に切れ込みがあり、これを目印にピン番号が決まっています。

規格表などに載っているICのピン番号はICの上から見たときのものなので注意してください。当然、配線している側から見ると逆回りになっています。この点は熟練者でも意外と勘違いすることがありますので油断しないように。もちろんこの連載では、実体配線図に従えばOKです。

次に、ICソケットを基板に差し込んだら、すぐに8番ピンと16番ピンを内側に折り込んでハンダ付けしてしまいます。というのも16ピンICの場合は8番がGND、16番が5Vに接続するのが一般的だからです。そして、基板がICB-87の場合は実体配線図を見てもわかるとおりICの2列の足の間にGNDラインと5Vラインの2本の配線ラインが通っているの、折り込んだ8番ピンと16番ピンとをそれぞれそのラインにもハンダ付けします。

このようにIC工作では、GNDラインと5Vラインを先に通してしまうのが基本なのです。これができればあとはICの足1本1本をすべてハンダ付けしていきます。今回は6番ピンもGNDに落とすので、内側に折り込んでGNDラインにハンダ付けします。

次に7セグメントLED (TLR313) をハンダ付けします。これはソケットがないので直接基板にハンダ付けしてしまうしかありません。TLR313のピン番号は先月号にも載せてありますが、やはりハンダ付けする側から見ると逆回りになっていることに注意しましょう。

TLR313の10番ピンは5Vラインに直結ですが、1～4、6、8、9番ピンは510Ωの抵抗を介してLS247につながるので、次に抵抗の配線を行うのが効率的です。配線の都合上、抵抗は実体配線図のように寝かして差し込み、TLR313側の足は折り曲げて、図4のようにTLR313の各端子まで伸ばしてハンダ付けしてやります。反対側は基板の端に並んでいる端子にハンダ付けしてやり、余った長さはすっぱり切り落としてしましましょう。

こうして7本の抵抗を付け終えたら、16進ロータリースイッチを取り付けます。私の手に入れたアルプス製のものは取り付け用の足も端子も位置としてはIC用基板に適したのですが、ただひとつ取り付け足が大きすぎて、基板の穴にはそのままでは入りません。そこで、錐(きり)を使って

取り付け位置の穴を少し大きくしてからはいれ込みます。はめ込んだら端子をハンダ付けしてしましましょう。

ここまでくると、部品はすべて取り付けられたことになります。ここでセンスの鋭い人はお気づきでしょうが、工作では、配線の前にすべての部品を取り付けてしまうのが鉄則です。それは何度もうのように、部品の配置とバランスが工作の手間を決めているといえるからです。

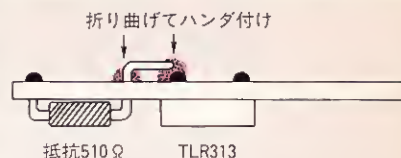
●部品の配線作業

部品がすべて配置されたら、次は地道に配線作業です。まずは抵抗7本とICとの接続をしましょう。TLR313のa～gから伸びている各抵抗の端をLS247のa～gに対応させて、被覆されたビニール線をつないでいきます。このようにつなぎたいところどうしをジャンプしてつないでいる線をことをジャンパ線といいます。

このとき、TLR313のa～gの並び方も間違えやすいですし、そのうえ、LS247のa～gが9～15番ピンに割り当てられていますが、これも順番に並んでいないので注意が必要です。実は、私も最初は間違えてつないでしまいました。間違えてつなぐとLEDの表示がおかしくなりますが、壊れることはありません。それから、10番ピンを5Vラインにつなぐのも忘れないように。以上で、LEDまわりの配線は完了です。

次にLS247の入力1、2、7番ピンの配線です。ここは、10ピンコネクタに直結しますが、10ピンコネクタのピン番号も間違いやすいので、再度図2を確認してくださ

図4 TLR313と抵抗のハンダ付け



い。このピンは位置も連載のすべての回路に共通です。

ところで、今回の製作でいちばん難しいのがこの10ピンコネクタまわりの配線でしょう。隣と近いうえ、ビニール線がかさばるので、次に述べるように手際よく行います。まず、コネクタ側の端子はあらかじめハンダ付けしておくこと。そして、ビニール線の被覆を必要な分(1mmほどで十分)だけワイヤストリッパでむいておき、そこにもハンダを付けておきます。

このように、ハンダ付けする両側にあらかじめハンダを付けておくのがコツです。あとは、ハンダゴテを基板側に当て、ビニール線の先をハンダ付けしたい箇所につけるだけで意外とうまくできます。万一隣にもくっついてしまった場合には、ハンダ吸い取り器で完全にハンダを取り除き、最初からやり直します。一度失敗したハンダは、二度とくっつかないことを肝に命じておく必要があります。

最後に3番ピンを5VラインにつないでICまわりの配線も終わりです。4、5番ピンはなにもしないでおきます。参考までに3～5番ピンの機能を囲み記事の中に記しておきますので、なにか自分で設計工作す

抵抗のカラーコード

抵抗1本1本をよく見ると色のついた4本の帯が見えますが、これが抵抗値を示すカラーコードです。精度の高い特別な抵抗には5本ついているものもありますが、ここでは一般によくみかける4本組の読み方を説明します。

最初の3本が抵抗値そのものを示し、最後の1本は抵抗の精度を示しています。精度というのは、表示されている値を基準にして実際の抵抗値にどれだけ誤差があるかということです。たとえばそれが金色のカラーコードであれば、実際の抵抗値は表示値の±5%という意味ですから、100Ωの抵抗の場合なら、実際は95～105Ωになっています。

最初の3本の見方をマスターしましょう。そこは0～9の9種類の色で表されていて、1本目と2本目とで2桁の値を示し、3本目でさらに10の何乗倍かを示します。図中の例題で確認してください。皆さんは、0～9が何色に対応するか覚えましょう。それには、0から9まですべて「くちあだき、みあむはし」と語呂で覚えます。それぞれの色名の頭文字を並べただけですが、なかなか覚えやすいと思います。

	1本目	2本目	3本目	4本目
黒(く)	0	0	10の0乗=1	
茶(ち)	1	1	1=10	±1%
赤(あ)	2	2	2=100	±2%
橙(だ)	3	3	3=1,000	
黄(き)	4	4	4=10,000	
緑(み)	5	5	5=100,000	
青(あ)	6	6	6=1,000,000	
紫(む)	7	7	7=10,000,000	
灰(は)	8	8	8=100,000,000	
白(し)	9	9	9=1,000,000,000	
金			10の-1乗=0.1	±5%
銀			-2=0.01	±10%

カラーコードの位置



	1	2	3	4	
例1	茶	黒	赤	金	1 0 × 10 ² = 1000Ω (1kΩ)
例2	緑	茶	茶	金	5 1 × 10 ¹ = 510Ω (今回使っているもの)

るときに参考にしてください。

あとは、10ピンコネクタの残りの端子を処理してやれば、完成です。そこでまず、16進ロータリースイッチとの配線を行います。ロータリースイッチには端子が5本出ているはずですが、1本は共通端子で、これはGNDラインにつなぎます。残りの4ビット端子はどの順に最下位ビットから並んでいるかあらかじめチェックしておいてから、10ピンコネクタの1～4番端子につなぎます。

店で品物を買うときに各端子の機能を尋ねておくのが得策です。自分で調べることになってしまったら、まずロータリースイッチを1に合わせておいて5本のうちの2本が導通しているかをテスターで計り、次に2、4、8と順次合わせて、やはりどの2本が導通しているかを調べます。1、2、4、8すべての場合に共通な端子がGNDにつながり、あとはそれぞれのビットに対応するかチェックします。

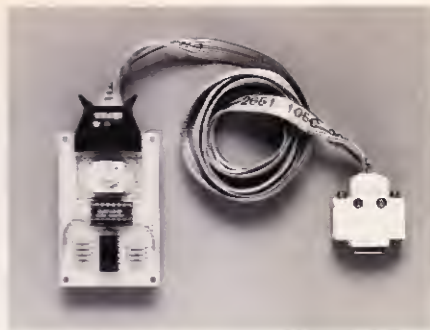
こうして、下位ビットから順に1～4ピンにつながればOKです。実体配線図では、下位ビットから順に1、2、4、8、COMと記号を打ってありますが、品物によって

位置が変わるかもしれません。最後に、10ピンコネクタの5番と5Vライン、9番とGNDラインとをつなぎます。この5VとGNDとを逆にするとICが死ぬこともあり得ますから、気をつけてください。

●完成後のチェック

以上ですべての配線が終了し、いよいよ完成です。配線が終わったら、実際にX68000につながり前にもう一度実体配線図と比べて、配線のチェックをしてください。ただし、一度配線が終わってからチェックするまでの間にはお茶を飲むなりゲームをするなり、なにか気分転換をすることが大切です。最初は必ずどこか配線ミスをしているものですが、これを発見するためには頭を冷やしたあとのほうがずっと効率が良いのです。

十分チェックしたら汎用ケーブルと基板のコネクタをつなぎます。コネクタには片側に出っ張りがあり、これで上下の向きが決まっていますので、向きに注意しながらしっかり差し込んで最後にフックで挟み込んで止めます。そしていよいよX68000のジョイスティックポート1に差し込んでみましょう。



これで完成だ！

どうですか、LEDに3が表示されましたか？ もし3が表示されなければ、まだどこかにミスがあります。ただし、このテストは必ずX68000の起動直後にX-BASICを立ち上げて行ってください。

とりあえず、あり得るミスについて考えてみましょう。

1) なにも表示されない場合

5VラインとGNDラインの配線ミスです。単にどこかの配線し忘れか、もしくは5VとGNDとを逆につないでしまっているかもしれません。

2) LEDは点灯するが、表示がおかしい

TLR313のa～gとLS247のa～gとの対応がきちんとできていない。あるいは隣どうしのピンがショートしていることもあり得ます。

3) 表示はするが、3でない場合

D+ピンコネクタが基板の10ピンコネクタまわりの配線ミス。LEDまわりの配線はOKです。

以上、どうしても配線ミスが見つからなければ、ICが死んでいることも考えられますが、実際のところICの不良は万に一つしかないと思って差し支えありません。根気よくミスを探してください。

*

いかがでしたか？ まったく初めて工作する人でもこの程度の回路なら十分ついて行けるのではないのでしょうか。完成したらさっそくX68000からコントロールしてみたいところですが、はやる気持ちを抑えて次回までのお楽しみとしましょう。

来月はまず、ソフトウェアで最も基本となるI/OコントロールドライバをX-BASICの外部関数の形で提供します。といってまたして睡しくないプログラムです。最初は68000アセンブラ入門みたいな解説になるでしょう。そのあとにそのドライバを使った応用プログラムを作ってみます。同時に一般的なI/Oコントロールを行うための基本もきっちり押さえる予定ですのでお楽しみに。ではまた、来月。

LS274の機能

図は規格表からの抜粋です。この図を見ながら各ピンの機能を順番に説明しましょう。

●電源系統

まず+5VとGNDは問題ないと思います。

●出力

出力（9～15番ピン）は7セグメントLEDのa～gに対応して、抵抗を介して接続します。図中に小さくa～gが書かれているのがわかるでしょうか。

●入力

入力は上位ビットからDCBAの順になっています。4ビット入力なので、0～15まで入力できますが、10以上になると意味のない表示になってしまいます。また、このICは表示の機能しかないので、たとえば桁上りを自動的に足し込むようなことはできません。

●オプション

3番ピンはランプテストといって、ここをGNDに落とすと強制的にすべてのセグメントを点灯させます。通常は5Vラインにつないでおきます。

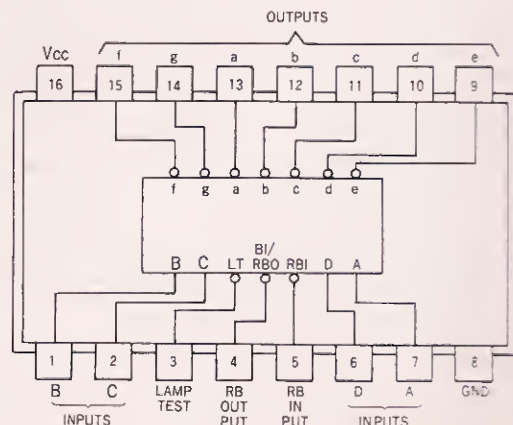
4番ピンのRBOと5番ピンのRBIはリップルブランキング機能に使うもので、通常はやはりなにもつながないでおきます。これはたとえば、4桁のLEDに3桁の数字を表示させるとき、上の桁の0を表示させないようにするために使います。

それには、最上位桁のRBIをGNDにおとし、そこから順に上の桁のRBOを次の桁のRBIにつないでいきます。RBI

がLのときは、もし入力が0ならばにも表示せず、しかもRBOからLを出します。

上の桁から順に0のときだけ数珠つなぎでなにも表示させないようにRBOをLにして伝達していきますが、ある桁で0でないときはそこから先はRBOがHのまま伝わっていくので途中の桁が0であっても0を表示します。

言葉で書くとなんとなくわかりづらいかもかもしれませんが、もし電卓や時計などで数桁にわたって表示させたいときにはそれぞれの用途で専用の表示用ICが簡単に手に入りますので、LS274のこの機能について理解しなくてもかまいません。それでも興味ある人は各自規格表を見て自由研究としてください。



超入門・ファイル処理

Izumi Daisuke 泉 大介

このところ難易度の高くなってきた調理実習ですが、今回は基本にかえて簡単なファイル処理の方法を解説しましょう。また、応用としてYETのスコアファイルを複数のプレイヤーで使用するためのアレンジも行っています。挑戦してください。

ゲーム作りもひと段落ついたところで、今月はちよつと実務っぽくファイル処理に取り組んでみたいと思います。一般にファイルというと、書類を綴じ込んだものを指しますが、コンピュータの世界ではディスクに保存されているBASICのプログラムやワープロの文書、表集計ソフトやデータベースのデータなどのことをいいます。文字やデータを綴じ込んだものだと見れば、なるほどファイルと呼ばれるのもうなづけるような気がします。この対比でいくと、ディスクドライブはさしずめファイルキャビネットというところでしょうか。

これらのファイルの内容に対するさまざまな作業がファイル処理で、簡単なところではファイルの中から単語を検索し、その単語が含まれている行だけを抜き出す。ファイルに入っている文字数、単語数、行数を数える、といった作業があります。ファイル処理というと難しそうな印象を持たれるかもしれませんが、コツをつかんでしまえば実に簡単なものなのです。なんせあのC言語では入門編で取り上げられる程度の題材なのですから。

ファイルのオープン、クローズ

ファイルを扱うときの儀式として、ファイルのオープン、クローズという作業があります。紙綴りファイルから必要な情報を探し出すときにはファイルを開きますね、また、作業が終わればファイルを閉じてキャビネットに戻します。これに対応するのがファイルのオープン/クローズです。ディスク上のファイルに開くも閉じるもないような気がしますが、コンピュータにとっては別の意味を持っています。

ファイルをオープンするとは、このファイルを使っているよとコンピュータに宣言する作業です¹⁾。これによってコンピュータはそのファイルが使用中であると認識し、ほかの人が同じファイルを使おうとするとエラーを出すことができるようになります。ファイルキャビネットなら使用中のファイルはキャビネット内にはありませんが、ディスクでは使っ

ていようがいまいが常にキャビネット内にファイルがあるようなものですからね。さらに進めて、見るだけなら何人の人が同時に見ようとディスク上のファイルが変更される心配はありませんから、複数の人がオープンできるようにすることもできます²⁾。

逆に、ファイルをクローズするというのは自分が使い終わったことをコンピュータに知らせるための作業です。X-BASICを終了するとオープンされたままのファイルは自動的にクローズされますが、自分でオープンしたファイルは必ず自分でクローズするようにしたいものです。

●見るのか、更新するのか、作るのか

ファイルを使うといってもいろいろあります。ファイルをオープンするときには、そのファイルを見るだけなのか書き込みをするのか、すでに存在するファイルを扱うのか新たに作るのかを明確にしなければなりません。これが「アクセスモード」あるいは単に「モード」と呼ばれるものです。

X-BASICでは「読む」「書く」「読み書きする」「新たに作る」という4つのモードでファイルをオープンすることができます。ワープロならば最初に文書ファイルを読むだけ読んで、変更が終わったあとに今度はすべて書き出せばOKですが、随時データの読み書きが行われるデータベースではファイルは「読み書き」モードでオープンする必要があります。メモリに入りきらないほどの大きなデータベースもあります。こうなると「読み書き」以外には扱う方法がありません。

●ファイル番号でファイルを管理

X-BASICでファイルをオープンするときには、
fopen (ファイル名, モード)
とします。モードはr (読む), w (書く), rw (読み書き), c (作る) と文字で指定するようになっています。たとえば、myfileというファイルを作りたいのなら、
fopen("myfile", "c")
となります。

ファイル処理を考えると、ファイルをひとつだけ

1) ファイル管理を一手に引き受けているのはOS(X68000ではHuman68k)ですので、ここは正確にはHuman68kに宣言する、となります。

2) マルチユーザーのOSではこの機能は必須といえるでしょう。X-BASICではファイルがすでにオープンされているかどうかのチェックすら行いませんが……。



しかオープンしないというのは稀です。あるファイルから特定の文字を探し出し、その文字列を含む行を別のファイルに書き出すというように、2つあるいは3つのファイルを同時にオープンして使うのが普通です。

ファイルがひとつだけならデータを読み込む、書き出す対象がどのファイルなのか迷うことはありません。しかし、オープンされているファイルが複数になると、対象がどのファイルなのか特定できなくなります。読み書きのたびにファイル名を指定するというのもひとつの解決法ですが、プログラムを書くのが面倒ですし、さらに1文字書くたびにファイル名の比較をやって対象のファイルを特定することになるので時間がかってしかたありません。

そこでファイル番号³⁾の登場です。オープンしたファイルに番号を付けておいて、あとはこの番号を利用して読み書きを行おうというものです。X-BASICではファイル番号はファイルをオープンしたときにfopen関数の戻り値として返される整数です。次の命令を試してみてください。

```
print fopen("test","c")
```

これでtestというファイルが新たに作成され、返されたファイル番号が画面に表示されるはずですが。ファイル作成を指定すると、すでに存在するtestというファイルを消去して新たに作ってしまいますので注意してください。実際にはこのファイル番号を変数に入れておきます。

```
int file
```

と宣言し、表示された値を代入しておきましょう。

続けてもうひとつファイルを作ってみます。

```
int file2
```

```
file2=fopen("test2","c")
```

変数file2を表示して、返されたファイル番号を確認してみましょう。

オープンしたあとのファイル操作はすべてファイル番号を使うと説明しました。ファイルのクローズも例外ではありません。クローズにはfclose関数を使い、引数にファイル番号を指定します。

```
fclose(file2)
```

なら、test2がクローズされます。もちろんtestはまだオープンされたままです。fclose関数はファイルを個々にクローズするのに便利な関数です。クローズ用の関数にはもうひとつfcloseallがあります。これはオープンされているファイルをすべてクローズするので楽なんです。では、次に進む前にfcloseall関数でファイルを全て閉じておくことにします。

```
fcloseall()
```

と入力すれば現在オープンされているファイルtestも(もしmyfileをオープンしているならそれも)クローズされます。

データを読み込んでみよう

さあ、いまや皆さんはファイルを開けたり閉じたりする方法を覚悟したわけです。fopen, fcloseという2つの関数はファイルの世界に入る最も基本的な呪文です。覚えての呪文はすぐに使って慣れるのがマジックポイント向上の秘訣とばかりに、さっそくfilesコマンドで表示されるファイルを片っ端から読み出しモードでオープンしている方もいらっしゃるでしょう。

そんな向上心旺盛なあなたに質問です。オープンできないファイルはありましたか? X-BASICの世界、すなわちHuman68kの世界にはこの方法でオープンできないファイルは存在しません。どんなファイルでも、それがfilesコマンドで表示されるファイルならオープンすることができるのです。ワープロの文書ファイルやBASICで作ったプログラムのファイルはもとより、皆さんが使っているX-BASICもオープン可能です。ワープロの文書やBASICのプログラムファイルは文字の集まりです。これに対しX-BASICはマシン語で書かれたBASIC本体です。X-BASICなどではこういったマシン語プログラムファイルはオープンすることができませんでしたが、X-BASIC(すなわちHuman68k)ではなんの制限もありません。文字が収められたファイルと実際にディスク上ではASCIIコードの集まりです。つまり1文字単位で読み込めば、0~255の数値が返ってくるだけなのです。ダンプリストでお馴染みのマシン語は16進数2桁(これも0

3) Human68kではファイルハンドルと呼んでいます。またファイルを指し示すものという意味でファイルポイントと呼ぶ場合も多々あります。ファイル番号を保持する変数にfpという名前が多いのはこのファイルポイントを略したものです。

4) X-BASICはBASIC.Xというファイル名でBASIC2ディレクトリ(あるいはBASICディレクトリ)に入っています。

～255の数値)の集まりですから、ファイル内では両者はまったく同じものだといえます。どんなファイルでもオープンできるというのはX-BASICのファイル処理の大きな特長です。

● 1文字単位で読み込む

では実際にファイルからデータを読み込んでみることにしましょう。まずは適当なプログラムを作り、それをTEST.BASというファイル名でセーブしてください。以前作ったプログラムがある方はそれを使って結構です。

まずはファイルのオープンです。データを読み込むのですから“r”でオープンします。

```
int file
file=fopen("TEST.BAS","r")
```

ですね。

さて1文字単位の読み込みですが、これにはfgetcという関数を使います。cはcharacterを意味しています。先ほど触れたように、この関数は文字を返すのではなく、ASCIIコードを返してきます。

```
print fgetc(file)
```

を実行してみてください。先ほど適当に作ってセーブしたプログラムの最初の文字のASCIIコードが表示されます。行番号の前にはスペースが詰まっていますから、スペースのASCIIコード32 (20H) が画面に表示されたはずですが、このままではわかりづらいので、chr\$関数でASCIIコードを文字に変換することにしてしましましょう。これは、

```
print chr$(fgetc(file))
```

でOKですね。ファイルの最後まで続けて表示するのなら、

```
while 1:print chr$(fgetc(file));:
endwhile
```

となります。セーブしたプログラムが表示され始めましたね。プログラムの最後まで表示すると……

「ピッ! (エラー音)」

ハイ、エラーです。

エラーが発生してしまいました (たぶん「バイトの範囲を越えました」と表示されているはず)。表示されたエラーメッセージを見てもなにが起こったのかわからないでしょうから解説しましょう。これはファイルの最後まで到達したにもかかわらず、さらにデータを読み込もうとしたのが原因です。ファイルの最後まで達すると、fgetc関数は-1を返します。

```
print fgetc(file)
```

として試してみましょう。ところがchr\$関数はchar型の引数(0～255)しか受け付けません。つまりchr\$(-1)を実行したのと同じことになりエラーが出たのです。ファイルを最後まで読み込んだら、それ

れ以上読みに行かないようにプログラムする必要があります。

ファイルの最後に到達したかどうかを調べるにはfeofという関数を使います。この関数は、

```
feof (ファイル番号)
```

という書式で利用し、指定されたファイルが最後まで(end of fileまで)達していたら-1を、まだ達していなかったら0を返します。これを使って、

```
while feof(file) <> -1 : ~ : endwhile
```

と先のwhileループを書き直せば、ファイルの最後まで文字を表示し続けることができます。fcloseall関数でTEST.BASファイルをクローズし、もう一度ファイルのオープンからトライしてみましょう。今度はエラーも起こりませんね。最後に、

```
fclose(file)
```

でTEST.BASをクローズすれば、ファイル処理入門はめでたく終了です。

リスト1はマシン語ファイルを表示するためのプログラムです。マシン語ファイルはchr\$で変換しても意味のある文字にはなりませんから、ダンプリストにならって2桁の16進数で表示することになりました。また数値がずらずらと並んでいるだけというのは見苦しいので、データ16個ごとに改行するようにしてあります。while～endwhileループでファイルエンドまで回しながら、for～nextを使って16個のデータを表示するという方法でプログラムしました。基本的には上の文字表示のプログラムと同じですからすぐにわかると思います。

このプログラムを使って、TEST.BASを表示してみましょう。2桁の16進数がずらずらと表示され、なにが入っているのかさっぱりわからないかもしれませんが、注意して見るとところどころに「0D 0A」というデータが入っているのがわかると思います。この2つのデータは改行を意味し、プログラムをロードするときX-BASICはこのデータを手掛かりに行の終わりを判定しているのです。

リスト1 マシン語ファイルを見る

```
10 str filename /* ファイル名
20 int file, data /* ファイル番号、データ
30 int readingFlag=1 /* 読み込み中フラグ
40 int i
50 /*
60 input "ファイル名: ", filename /* ファイル名入力
70 file=fopen( filename, "r" ) /* ファイルオープン
80 while readingFlag /* 読み込み中は以下を実行
90   for i=1 to 16 /* 16回繰り返す
100     if feof( file ) = -1 then {
110       readingFlag = 0 /* ファイルエンドならフラグ
120       break /* をクリアしてループ中断
130     }
140     data=fgetc( file ) /* データを1つ読み込み
150     print hexStr( data ); " "; /* 16進で表示
160   next
165   print /* 次の行へ
170 endwhile
180 fclose( file ) /* ファイルを閉じて
190 end /* 終了
200 /*
210 func str hexStr( data ) /* 16進2桁の文字にする
220   return( right$( "0"+hex$(data), 2 ) )
230 endfunc
```


の2つの配列を用意し、これにscoFileのデータを取り出してセットします。

プレイヤーの名前は必ず6文字分としてscoFileに収めてあり、そのあとにスコアがセットしてありますから、

```
for i=0 to 9
  for j=0 to 5
    player(i)=player(i)
    +chr$(scoFile(i*7+j))
  next
  score(i)=scoFile(i*7+6)
next
```

として2重ループを作ればplayerとscoreの2つの配列にデータをセットすることができます。6つの名前データと1つのスコアデータが1組になっていますから、i*7番目から6つのデータを取り出しそれを文字列に変更してplayer配列のi番目に、その次のデータを取り出してscore配列のi番目にセットしているのが上のプログラムです。

ではここでプログラムを見ていただきましょう。リスト2です。10行ではスコアファイル名をユーザが設定できるように変数として宣言しています。20行はいま説明したデータ読み込み用配列、そして30、40行がplayer配列とscore配列です。ここでは3つ宣言してありますね。これは、このあと同じ名前の削除を行うのに、ひとつの配列の中でやりくりするのは面倒なためです。加工後のデータは別の配列に入れることにしました。

上で説明したファイル読み込みおよびplayer、score配列へのセットを行っているのは1160行のreadSco関数です。ここでは引数nの値によって、player1、score1にセットするのか、player2、score2にセットするのかを振り分けています。

●同一人物の削除

同一人物を削除するには、同じ名前を飛ばしてスコア配列を詰めていけばOKです。readSco(1)でplayer1、score1配列にデータを読み込み、player1配列を上から順に見ていって、初めて登場する名前なら名前とスコアをplayer、score配列へ移します。

問題は初めて登場する名前かどうかを判定する方法です。player配列を順に調べてもいいのですが、ここではinstr関数を使うことにしました。instr関数は、文字列が特定の文字列を含んでいるかどうかを判定する関数です。player1配列からplayer配列へ移した名前を文字型変数chkStrに順次代入していくことにすれば、ある名前をplayer配列に移したかどうかはchkStrを調べるだけですみます。

ここで気をつけなければならないのは、文字列を単純に追加してはいけないということです。スコア

のトップがdai、2番目がdanだったとします。単純に追加するとchkStrは、

daidan

となりますね。スコアの3番目がidaだとすると、idaはすでにchkStrに入っていることになってしまうため、player配列へ移されません。

このような事態を避けるため、名前の前後を決して名前に使われない文字で区切る必要があります。決して名前に使われない文字を仮に‘.’だとすると、chkStrは、

.dai.dan.

となり、“.ida.”はこの中に含まれないのでうまくいきます。

これらの処理を行っているのが320行から始まるunify関数です。ここでは区切り文字としてchr\$(1)を使っています。

●整頓後のスコアの保存

スコアの保存はスコアの読み込みと逆の手順で行います。整頓が終わったスコアはplayer、score配列に収められていますから、これら2つの配列からscoFile配列へデータを移し、それをfwrite関数で一気に書き出せばOKです。これは1570行のsaveSco関数が行っています。

●2つのスコアファイルを融合する

player2、score2配列が用意してあるのは、この機能を実現するためです。1つ目のファイルをplayer1、score1配列に、2つ目のファイルをplayer2、score2配列にセットし、これら2つの配列から点数の大きいものの順にplayer、score配列へと移していくと融合が完成します。具体的には2つの配列の添字用に2つの変数(rank1、rank2)を用意し、

```
if score1(rank1)>=score2(rank2) then {
  配列 player1, score1をplayer, scoreへ
  rank1=rank1+1
} else {
  配列 player2, score2をplayer, scoreへ
  rank2=rank2+1
}
```

とします。添字変数はデータを移したときだけ大きくなり、次のスコアがもう一方のスコアと比較されることになります。

この処理を行っているのが730行から始まるmerge関数です。画面表示処理が間に入っているので若干わかりづらいかもしれませんが、やっていることは上で説明したことだけです。

●プログラムの拡張について

さて毎度のことながら、プログラムには必要最小限の機能しか盛り込んでありません。エラー処理はまったくやっていませんし(X-BASICで実行する

システムディスクのBINディレクトリにはDUMP.Xというプログラムが入っています。これは

```
69 6E 70 75 74 20 22 ..... input " .....
```

というように、16進数とそれをASCIIコードと見なしたときの対応する文字を表示してくれます。リスト1はこの左半分だけを表示するようなものです。リスト1を改造し、DUMP.Xのような出力ができるように挑戦してみてください。

●データを読み込むそのほかの関数たち

X-BASICではfgetcのほか、freads, freadの2つの関数でデータをファイルから読み込むことができます。freadsは文字が入っているファイルを対象とし、改行コードまでの1行を一気に文字変数に読み込む関数です。1文字1文字読み込むより一気に読むほうが速いので、文字ファイル処理では多用される関数です。もう一方のfreadは1次元の数値型配列を一気にファイルから読み込む関数です。

●データを書き出す

ファイルにデータを書き出すときには、“w”モードか“c”モードでファイルをオープンし、データ書き出し用の関数を使うだけで基本的な作業はまったく同じです。データ書き出し用に用意されている関数はfputc, fwrites, fwriteの3つで、これまでに紹介してきた読み込み用関数と対になっています。

fwriteは実験データなどを1次元の配列に収めておき、「ハイ、セーブ!」と一発で処理できる便利な関数です。

YET再び

6月号付録ディスクのYET.Xはトップ10のスコアをファイルに残します。もともとオマケ的な要素が強かったので暗号化も行わず、単純に名前とスコアを記録するようになっています。DUMP.Xで覗くとその構造がよくわかるでしょう。作成当時には最高得点は3万点台が限界だろうと思い、このあたりなら十分自分の名前を残すことができるという自負から、同じ人物の得点は最高点のみを残すなどという細工を行わなかったのです。

ああそれなのに、それなのに。編集室ではいつしかトップ10すべてが4万点台になってしまったのです。しかもたった2人の人物によって! 結局私はやってもやってもスコアを残すことができず、「これはなんとかしなければ」という使命感のもと、スコア調整プログラムを作ることになりました。

このスコア調整プログラムは次の2つの機能を持っています。

- 1) 同じ人物のスコアは最高得点のみを残す
- 2) 2つのスコアファイルを融合する

1)は1人の人物がスコアを独占し、ほかの人が名前を登録する荣誉にあずかれないという事態を打破するために用意しました。2)は自宅でさんざんやって出した高得点をクラブのX68000に移し、友達に尊敬されるためです。

●YETSCOのファイル構造

YETのスコアファイルであるYETSCOは整数型の配列をfwrite関数でファイルに書き出しただけの非常に簡単な構造をしています。1人分のデータは、

1. 名前の1文字目のASCIIコード
2. 名前の2文字目のASCIIコード
- ...
6. 名前の6文字目のASCIIコード
7. スコア

という形式で7つの整数型データに変換され、これが10人分続いたのがスコアファイルなのです。例をお見せしましょう。「DAISKE 32000」というスコアをこの方法で変換すると、

```
68 65 73 83 75 69 32000
```

となります。

●まずはスコアファイルの読み込みから

ではまず、スコアファイルの読み込みです。スコアファイルはfwriteで書き出したファイルですので、読み込みはfreadで行いましょう。1人のデータが整数7個分ですから、10人のデータは整数70個分になります。

```
int scoFile(70)
```

でデータを読み込む1次元配列を作成し、

```
int file  
file=fopen("yetsco","r")  
fread(scoFile, 70, file)  
fclose(file)
```

でデータの読み込みは終了です。freadは読み込む配列名と、読み込むデータの個数、そしてファイル番号を引数にとります。

●名前とスコアを取り出す

データをいったん読み込んでしまえば、あとは普段のプログラミングと変わりありません。これまで初期値を与えた配列を使うプログラムをいくつか作ってきましたが、初期値を与える代わりにファイルから読み込んだだけだと考えてもいいでしょう。

いま、(ファイルから読み込んで)初期値を与えた配列scoFileがあります。これはASCIIコードとスコアをごちゃまぜにして登録してある配列です。このままでは扱いづらいので、プレイヤーの名前を入れた配列と、スコアを入れた配列に分けることにします。

```
str player(9)  
int score(9)
```


なら、致命的なエラーはBASICが出してくれる)、処理を途中でやめなくなった場合のことも考慮してありません。整頓終了後に画面に表示される結果が気に入らない場合は、ファイル名入力のプロンプトが表示されているときにブレイクしてください。

まず最初に皆さんに取り組んでもらいたい拡張は、2つのファイルを融合するときに同一人物を削除する機能を付加することです。unify関数が参考になるかと思います。

リスト2 YETのスコア管理ぶるぐらむ

```

10 str scoName          /* スコアファイル名用
20 int scoFile(70)      /* データ読み込み用1次元配列
30 str player(9), player1(9), player2(9)
40 int score(9), score1(9), score2(9)
50 int conFlag = 1
60 str selection
70 /*
80 while conFlag
90   cls
100  print "Score Manager"
110  print
120  print "1) 同一人物削除"
130  print "2) ファイル融合"
140  print "3) 終了"
150  print
160  print "処理する番号:";
170  selection = inkey$
180  switch selection
190    case "1"
200      unify()
210      break
220    case "2"
230      merge()
240      break
250    case "3"
260      conFlag = 0
270      break
280  endswhile
290 endwhile
300 end
310 /*
320 func unify()
330  str chkStr(80)
340  int i, rank=0
350  /*
360  /* スコアファイル読み込み
370  /*
380  locate 0, 10
390  input "スコアファイル名:", scoName
400  readSco( 1 )
410  cls
420  for i=0 to 9
430    print player1(i), score1(i)
440  next
450  print
460  /*
470  /* 同一人物削除
480  /*
490  print "  変換"
500  print
510  chkStr = chr$(1)
520  for i=0 to 9
530    player(i) = ""
540    score(i) = 0
550    if instr( 1, chkStr, chr$(1)+player1(i)+chr$(1) ) = 0 then
560      chkStr = chkStr + player1(i) + chr$(1)
570      player(rank) = player1(i)
580      score(rank) = score1(i)
590      rank=rank+1
600    )
610  next
620  for i=0 to 9
630    print player(i), score(i)
640  next
650  print
660  /*
670  /* スコアファイル保存
680  /*
690  input "セーブします。ファイル名:", scoName
700  saveSco()
710 endfunc
720 /*
730 func merge()
740  int rank, rank1, rank2
750  /*
760  /* 2つのスコアファイル読み込み
770  /*
780  locate 0, 10
790  input "スコアファイル名1:", scoName
800  readSco( 1 )
810  input "スコアファイル名2:", scoName
820  readSco( 2 )
830  cls
840  for i=0 to 9
850    print player1(i), score1(i), player2(i), score2(i)
860  next
870  print
880  /*
890  /* 2つのスコアを1つにまとめる

```

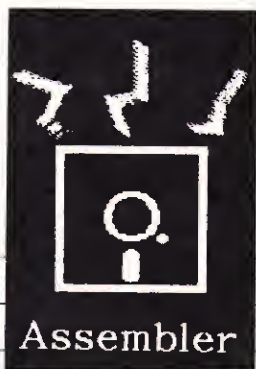
yetscoファイルを読み込み変数にセットすることのプログラムを使えば、簡単にスコアを変更することができてしまいます。player、score配列の中身を適当にいじってsaveSco関数を呼び出すだけでいいのですから5万、10万点のスコアなんて楽勝です。そんなスコアを見せびらかして喜ぶような悲しい遊びはやらないでくださいね。

来月は「ちょっと高度なファイル処理」と称してデータベースもどきをお送りする予定です。

```

900  /*
910  print "  変換"
920  print
930  rank1=0 : rank1=-
940  for rank=0 to 9
950    if score1(rank) < score2(rank) then {
960      player1(rank) = player1(rank)
970      score1(rank) = score1(rank) + score2(rank)
980      rank1 = rank + 1
990    } else {
1000     player1(rank) = player1(rank)
1010     score1(rank) = score1(rank) + score2(rank)
1020     rank2 = rank + 1
1030   }
1040 next
1050 for i=0 to 9
1060   print player1(i), score1(i)
1070 next
1080 print
1090 /*
1100 /* スコアファイル保存
1110 /*
1120 input "セーブします。ファイル名:", scoName
1130 saveSco()
1140 endfunc
1150 /*
1160 func readSco( n )
1170  int file
1180  int i
1190  str ch
1200  /*
1210  /* 対象とするスコア配列にデータ読み込み
1220  /*
1230  file = fopen( scoName, "r" )
1240  fread( scoFile, 70, 1, file )
1250  fclose( file )
1260  /*
1270  /* 対象とする名前配列を初期化
1280  /*
1290  for i=0 to 9
1300    if n = 1 then {
1310      player1(i) = ""
1320    } else {
1330      player2(i) = ""
1340    }
1350  next
1360  /*
1370  /* 名前配列に名前を
1380  /* スコア配列にスコアをセット
1390  /*
1400  for i=0 to 9
1410    for j=0 to 5
1420      ch = chr$( scoFile( i*7 + j ) )
1430      if n = 1 then {
1440        player1(i) = player1(i) + ch
1450      } else {
1460        player2(i) = player2(i) + ch
1470      }
1480    next
1490    if n = 1 then {
1500      score1(i) = scoFile( i*7 + 6 )
1510    } else {
1520      score2(i) = scoFile( i*7 + 6 )
1530    }
1540  next
1550 endfunc
1560 /*
1570 func saveSco()
1580  int file
1590  int i
1600  /*
1610  /* 名前配列から名前を
1620  /* スコア配列からスコアを取り出し
1630  /* scoFileにセット
1640  /*
1650  for i=0 to 9
1660    for j=0 to 5
1670      scoFile( i*7 + j ) = asc( mid$( player(i), j+1, 1 ) )
1680    next
1690    scoFile( i*7 + 6 ) = score( i )
1700  next
1710  /*
1720  /* scoFileを書き出す
1730  /*
1740  file = fopen( scoName, "w" )
1750  fwrite( scoFile, 70, 1, file )
1760  fclose( file )
1770 endfunc

```

マウスwithグラフィック

Murata Toshiyuki 村田 敏幸

X68000用のプログラムを作成するというのなら、やはり、マウスも基本として押さえておきたいところ。マウス制御のためのさまざまな機能がIOCSとして用意されていますからこれを利用するのが正攻法です。簡単なお絵かきツールで実践してみましょう。

最初に、前回の記事中にボカがあったので訂正しておく。アドレスレジスタにaddqやsubqで小さな定数を加減算するときにはワードサイズを指定したほうが速いと言ったが、大嘘なので忘れてほしい。実際には、ロングワードでもワードでも実行速度は変わらない。また、最後のASX.Sの中で使っているインクルードファイルが抜けていた。リスト0にそのFILES.Hを示す。ひと月休んで訂正が遅れたことと合わせてお詫びする。痛惜の念に堪えない、ぐらいのことはいうべきなのかもしれないが、この言葉はいつかともない大バグを出したときのためにとっておこうと思う。

*

さて、今回は地味ながらX68000らしいところでマウスを取り上げ、最後はこれにパラパラとグラフィックを振りかけてこぢんまりとまとめてみたい。あくまでマウスがメインであり、グラフィックまわりについてはあまり詳しく触れないことをあらかじめ断っておく。

IOCSコールを使う

X68000ではROMにIOCS (Input/Output Control System)の形でさまざまな機能の制御ルーチンが用意されており、マウスもこのIOCSを呼び出すことによってほとんどX-BASICと同じ感覚で手軽に利用することができる。もうご存じだとは思いますが、一応、IOCSの概要と呼び出し手順を押さえておこう。

X68000のIOCSはテキスト画面への文字表示、キー入力に始まって、プリンタ出力、RS-232Cによる入出力、フロッピーディスク/ハードディスクの物理的な読み書き、マウスの制御、グラフィック描画、スプライト、AD PCM、カレンダー時計などの周辺LSIの制御にいたるまで、X68000の(ほとんど)すべての機能を網羅している。位置づけとしてはシステム中もっともハードに近い部分を担当しX68000上のプログラムを底辺からささえる低レベルI/Oルーチン集¹⁾であり、OSであるHuman68kもIOCSに乗った形で作られている。

これにより、ユーザープログラムがHuman68kに入出力を要求すると、Human68kは必要に応じてIOCSを呼び出し、最終的にIOCSがハードに働きかけて物理的な入出力を行う²⁾。結果は逆のルートを伝って返される。このHuman68kとIOCSとの上下関係(というか依存関係というか階層構造というか)は心に留めておいてもらいたい。

IOCSを呼び出す手順はいたって簡単で、d0,1にIOCSコール番号を入れてtrap #15という命令を実行するだけだ。パラメータがあるときはd1以下のデータレジスタやa1以下のアドレスレジスタ(a0はIOCSコール呼び出しには使われない)に入れて渡す。たとえば、IOCSコール番号20_Hに割り当てられている1文字表示機能を使うときには、

```
move.w    #'A',d1
moveq.l   #$20,d0
trap      #15
```

IOCSコール番号21_Hの文字列表示機能を使うのなら、

```
lea.l     mes,a1
moveq.l   #$21,d0
trap      #15
:
```

1) 実際にはハードがらみ以外にも、シフトJIS漢字コード↔JIS漢字コードの相互変換とか、ユーザーモードからスーパーバイザ空間にあるメモリを読み書きするといったユーティリティ的なものもIOCSには用意されている。

2) 論理的には、OSの低レベルI/Oはデバイスドライバが担当することになっているわけだが、現実にはHuman68kのデバイスドライバはさらに下位の存在であるIOCSを下請けに使っている場合が多い。

trap命令

trapは端的にいうと故意に例外を発生させる命令だ。trapにはtrap #0~#15の16個があり、順に例外ベクタ番号20_H~2F_H、例外ベクタアドレスでいうと0080_H以降の16ロングワードが割り当てられている。

trap命令が実行されると68000はスーパーバイザモードに移行し、命令が実行された時点でのpcとsrの値をスーパーバイザスタックに積む。そののち、該当する例外ベクタの内容を参照し、指定されたアドレスから例外処理を実行する。DOSコールの呼び出しに利用されている未実装命令の実行による例外とは異なり、trap命令による例外処理開始時にスタックに積まれるpcは命令が置かれた直後のアドレスを指しており、小細工をしなく

とも例外処理の最後でrteを実行すればtrap命令のすぐうしろからプログラムの実行を再開できる。

感覚としては“スーパーバイザモードへの移行を伴うサブルーチンコール命令”といったようなもので、その性質上、システムコールを呼び出すのによく用いられている(そのようにシステムが設計される)。本文でも触れたようにX68000ではIOCSの呼び出しにtrap #15を使っている。

このほかX68000+Human68kではtrap #8~#14を内部的に使用している。ふつうのプログラムを作るうえでは知らなくてもすむのだが、興味のある人は「プログラマーズマニュアル」の3.2節末にある参考資料を見てみるとよいだろう。

mes: .dc.b '文字列',0
 という具合だ。ソースプログラム中にIOCSコール番号を生のまま埋め込むのがいやであれば、Human68kのDOSコールの場合のように、インクルードファイルをひとつ作成してその中で、

```
__B__KEYINP      equ    $00
:
__B__PUTC        equ    $20
__B__PRINT       equ    $21
:
```

のようにずらずらとIOCSコール番号をシンボル定義しておけばよい。幸いなことにXCにはこのインクルードファイルがIOCSCALL.MACの名前であらかじめ用意されている。また、IOCSCALL.MAC内では、

```
IOCS macro callno
    moveq.l #callno,d0
    trap #15
endm
```

というマクロが定義されていて、このマクロとシンボルを利用すると上の例は、

```
move.w #'A',d1
IOCS __B__PUTC
とか、
lea.l mes,a1
IOCS __B__PRINT
```

のようにすっきり書けるようになる。今後この連載でIOCSコールを利用するときにはIOCSCALL.MACをインクルードし、このスタイルで記述する(編集部注:本誌6月号の付録ディスクにも収録させて

いただいたので利用してください)。

実際にIOCSコールを使ったプログラムの一例をリスト1に示す。こんな機会でもなければ誌面に載ることもないようなちっぽけなプログラムLEDOFF.F.Xだ。実行するとすべてのLEDキーをOFF状態にする。AUTOEXEC.BATに潜り込ませるか、Human68k Ver.2ならCONFIG.SYSのPROGRAM=～行に記述するかして起動時に1回走らせるのが正しい使い方だ。起動直後に“ち”とか“ぢ”と打ち込んで“コマンドまたはファイル名が違います”攻撃を受けたことがある人ならLEDOFF.Xの有用さに気づいてもらえると思う。

見てのとおりプログラムはLEDキーの状態を操作するIOCSコールLEDMODをループの中から実行するだけという単純さだ。LEDMODは2つのパラメータを取り、d1,lでLEDキーの番号(0～6)、d2,bでONにする(1)かOFFにする(0)かを指定する。リスト1ではループ内でd2を0に固定したままd1を順に変化させてすべてのLEDキーをOFFにしている。このことから察しがつくと思うが、IOCSコールでは基本的にd0以外のレジスタの値は保存される(例外はある)。d0だけはIOCSコールの終了ステータスないしは適当な戻り値を返すのに使われる。ちなみに、LEDMODはパラメータの値が範囲外でLEDの設定できなかった場合は-1を、うまく設定できたときは0をd0,lに返す。

そしてマウスへ

とんとんとマウスの話に進む。マウス関連のIOCSコールはコール番号70h～7Dhにまとめられており、『プロデューサズマニュアル』を見てもらえばわかるように、X-BASICのマウス操作関数と似たような機能を持ったものがずらっと並んでいる。

X-BASICでマウスを使ったことがあれば、これらを使いこなすものではない。さっさとサンプルにいてしまっただけで済まないだろう。リスト2のMTEST.Sは画面にメニューをひとつ表示し(実際には“終了”という文字列を左上隅に書くだけ)、このメニューの上で左ボタンが押されたら、それに応じた処理をする(終了する)プログラムだ。

11～14行はマウスを使うときには枕詞のように現れる初期化・使用準備の決まりきった手順だ。最初のMS__INITによりマウスカーソルの表示はOFFになり、カーソルパターンは標準の矢印型に、カーソル座標は(0,0)に、カーソルの移動範囲は表示画面の大きさと一致するように初期化される。つづくMS__CURONでマウスカーソルを表示状態にし、SKEY__MODでマウスの右ボタンに割り当てられているソフトウェアキーボードとマウスカーソルの表示・非表示切り換え機能を殺して初期化は完了だ。この3つのIOCSコールの組み合わせは、X-BASICの

```
mouse (0)
mouse (1)
```

リスト0 FILS.H

```
1: *      nameck,files,nfiles用オフセット定義
2: *
3: *      .offset 0
4: *
5: DRIVE: .ds.b 2      *ドライブ名 'A:'
6: PATH:  .ds.b 64+1   *パス名 'BIN\',0
7: NAME:  .ds.b 18+1   *ファイル名 'ATTRIB',0
8: EXT:   .ds.b 1+3+1  *拡張子 '.X',0
9: *      .even
10: NMBUFSIZ:
11: *
12: *      .offset 0
13: *
14: FORSYS: .ds.b 21     *システムが使用
15: FATR:  .ds.b 1      *ファイル属性
16: FTIME: .ds.w 1      *ファイル最終更新時刻
17: FDATE: .ds.w 1      *ファイル最終更新日
18: FLEN:  .ds.l 1      *ファイル長
19: PACKEDNAME:
20: *      .ds.b 18+1+3+1
21: *      .even
22: FJLBUFSIZ:
23: *
24: *      .text
```

リスト1 LEDOFF.S

```
1: *      全てのLEDキーをOFFにする
2: *
3: *      .include      iocscall.mac
4: *      .include      doscall.mac
5: *
6: ent:
7: *      moveq.l #0,d2      *OFF
8: *      moveq.l #7-1,d1    *LEDキー番号
9: loop: IOCS __LEDMOD      *設定
10: *      dbra d1,loop      *繰り返し
11:
12: DOS __EXIT              *終了
13:
14: .end ent
```


mouse (4)

にほぼ対応している。

16行からメイン処理が始まる。まず、左ボタンが押されるまで待つ (16~18行)。ボタンの状態を得るにはIOCSコールMS_GETDTを利用する。このIOCSコールはX-BASICのmsstat () に相当し、d0,lの上位ワードにマウスカーソルの相対的な移動量を、下位ワードに左右のボタンの状態を返す。相対的なカーソル移動量のほうはあまり利用されることはないはずだからここでは触れない。

ボタンの状態は第0~7ビットに右ボタン、第8~15ビットに左ボタンのON/OFF状態が返り、ボタンが押されているときは8ビットとも1 (FF_H)、押されていない場合は8ビットとも0 (00_H) になる。右ボタンが押されているかどうかチェックしたければ、

```
IOCS  _MS_GETDT
```

```
tst.b  d0
```

```
beq   押されていない
```

```
       押されている~
```

のようにtst.b後のZビットで処理を振り分ければよいのは明らかだろう。左ボタンの場合は、

```
IOCS  _MS_GETDT
```

```
tst.w  d0
```

```
bpl    押されていない
```

```
       押されている~
```

という手が使える。どうせ8ビットとも同じ値をとるのだから、第15ビットだけを調べればすむわけだ。

MS_GETDTで左ボタンの押し下げが検出されたら、すかさずMS_CURGTでマウスカーソルの画面上での現在位置を得る (21行)。MS_CURGTはX-BASICのmspos () 関数に当たり、d0,lの上位ワードにマウスカーソルのX座標、下位ワードにY座標を返す。得られた座標がメニュー上にあるかどうかを調べているのが25~28行、やっているのは単純な座標の比較だ。

最後に31行以下が忘れてはならない後始末の処理だ。MS_INITでマウスを再初期化して(マウスカーソルを消し)、SKEY_MODでさっき殺したソフトウェアキーボードを使用可能状態に戻している。

マウスについてはだいたいリスト2の応用で片がつく。あと、ダブルクリックの判定方法ぐらいは知っていたほうがいいのかも。そこでリスト3。リスト2の30行以下と差し換えて使う。ダブルクリックの判定といってもやるべきことは泥臭いといっているほど直接的だ。ボタンが押されたことがわかったら、

1) 一定時間以内に離されるかどうか

2) 一定時間以内にまた押されるかどうか

というチェックを続けて行い、両方に通ったらダブルクリックされたと判断する。これには、IOCSコールのMS_OFTM、MS_ONTMを利用する。d1,wで左右のボタンのどちらか (0なら左、-1なら右)、d2,wで待ち時間を指定し (とくに0のときは

無限と見なされる)、指定時間内にボタンが離されたり (MS_OFTM) 押されたり (MS_ONTM) したら、それまでの経過時間をd0,wに返す。ただし、ドラッグされた場合 (ボタンの状態が変化しないうちにマウスカーソルが動いた場合) にはd0,w=0で即戻ってくる。また、待ち時間を越えた場合はFFFF_Hが返る。待ち時間の単位はなにやらないか加減らしく (ループ回数で計時しているのかな)、だいたい40が0.1秒前後に相当する。リスト3では待ち時間を0.2秒程度にするために80を指定してある。

リスト2 MSTEST.S

```
1:      .include      iocscall.mac
2:      .include      doscall.mac
3:      .include      const.h
4:      *
5:  ent:
6:      lea.l    mysp(pc),sp
7:
8:      lea.l    menu(pc),a1
9:      IOCS     _B_PRINT      *メニューを描く
10:
11:      IOCS     _MS_INIT      *マウス初期化
12:      IOCS     _MS_CURON     *マウスカーソル表示
13:      moveq.l  #0,d1         *ソフトウェアキーボード
14:      IOCS     _SKEY_MOD     *表示禁止
15:
16:  loop:  IOCS     _MS_GETDT    *ボタンの状態を得る
17:         tst.w  d0            *左ボタンは押されているか?
18:         bpl    loop         *押されていない
19:
20:         *左ボタンが押された
21:      IOCS     _MS_CURGT     *マウスカーソル座標を得る
22:      move.w   d0,d1         *d1.w = Y座標
23:      swap.w   d0            *d0.w = X座標
24:
25:      cmpi.w   #32,d0        *X座標のチェック
26:      bcc      loop         *範囲外
27:      cmpi.w   #16,d1        *Y座標のチェック
28:      bcc      loop         *範囲外
29:
30:         *終了メニュー上だった
31:      IOCS     _MS_INIT      *マウス再初期化
32:      moveq.l  #-1,d1        *ソフトウェアキーボード
33:      IOCS     _SKEY_MOD     *表示許可
34:
35:      DOS      _EXIT         *終了
36:      *
37:      .data
38:      .even
39:      *
40:  menu:  .dc.b   26,'終了',CR,LF,0
41:      *
42:      .stack
43:      .even
44:      *
45:  mystack:
46:      .ds.l    256
47:  mysp:   .end    ent
```

リスト3 MSTEST2.S

```
30:         *終了メニュー上だった
31:      moveq.l  #0,d1         *左ボタン
32:      moveq.l  #80,d2        *待ち時間 (約0.2秒)
33:      IOCS     _MS_OFFTM     *離されるまで待つ
34:      tst.w    d0            *0以下なら
35:      ble      loop         *はしく
36:
37:      IOCS     _MS_ONTM     *押されるまで待つ
38:      tst.w    d0            *0以下なら
39:      ble      loop         *はしく
40:
41:         *ダブルクリックされた
42:      IOCS     _MS_INIT      *マウス再初期化
43:      moveq.l  #-1,d1        *ソフトウェアキーボード
44:      IOCS     _SKEY_MOD     *表示許可
45:
46:      DOS      _EXIT         *終了
47:      *
48:      .data
49:      .even
50:      *
51:  menu:  .dc.b   26,'終了',CR,LF,0
52:      *
53:      .stack
54:      .even
55:      *
56:  mystack:
57:      .ds.l    256
58:  mysp:   .end    ent
```


お絵かきツールへの応用

最後に応用プログラムとして、簡単なお絵かきツール（グラフィックエディタなんて呼べるほどの代物ではない）を作って今月はおしまいにする。当初はマウスボタンが押されたらその位置に点を打つだけのプログラムにしようと思っていたが、これだとあまりに単純すぎて面白みに欠けるので、IOCSコールで実現できる範囲で多少彩りを添えてみた。

- 1) 色の選択は右ボタンを押すことでポップアップするウィンドウで選べるようにする
- 2) 同じウィンドウ上にはペンパターンメニューも並べ、複数の中からペンのパターンを選べるようにする（パターンは最大16×16ドット）

一見複雑な処理が要求されそうだが、X68000のハードの機能とIOCSのおかげで、どちらも簡単に実現できる。まず、ウィンドウをポップアップする処理だが、256色2画面の画面モードを使用して、1画面をウィンドウ用、残りを描画用と使い分けることで逃げた。ウィンドウはあらかじめ全部描いておき、X-BASICのvpage関数、home関数に相当するIOCSコールVPAGEとHOMEで表示のON/OFF、表示位置の変更を行う。2点目のペンパターンについては、“外字をSYMBOLで表示する”という手を使った。ペンのパターンを外字に登録しておき、PSETで点を打つ代わりにSYMBOLで描くわけだ。どちらもかなり安直だが、彩りとしての役目は果たしてくれる。

グラフィック関係のIOCSについては約束どおり特に解説しないから『プログラマーズマニュアル』を参照してもらいたい。一応リスト4にLINEのサンプルを示しておく。COMMAND.X上からグラフィック画面に直線を描画するプログラムだ。7行のパラメータの個数と、23行のIOCSコール番号を変更すればBOXやFILL、CIRCLEにも対応できるので気が向いたら試してみしてほしい。あまり使い道のないプログラムだが、派手なパッチファイルを作りたいときなんかには利用できるだろう。

なお、コマンドラインで指定された数字（の文字列）を数値に変換するのにリスト5中のサブルーチンatoiを利用しているので、実行ファイル作成時にはこれも忘れずにリンクすること。このatoiは今後使うことがあるかもしれない（変に凝ってしまったのであまりよいではないが）。また、LINE.Xはグラフィック画面の初期化を行わないので、使用時にはSCREENコマンドであらかじめグラフィック画面を使用可能に設定しておく必要がある。

atoiについて1点だけ補足しておく。5～8行ではCフラグを反転（0 ↔ 1）するマクロCCFを定義している。その実体は、

```
eorl.w #1, ccr
```

というオペランドにccrが登場するという見慣れない命令だ。この命令は任意のフラグを反転するのに

使う。排他的論理和の意味と、ccrの構造を思い出してもらいたい。同様の命令としては、

```
andi.w #n, ccr
```

```
ori.w #n, ccr
```

があり、それぞれ、ccrレジスタ中の任意のフラグをリセットしたりセットしたりするのに用いられる。

STAMP.Sの解説

では、手抜きいっぱいのお絵かきプログラム、リスト6のSTAMP.Sを見てもらおう。比較的読みやすく書けたと思うので、これまでの話のまとめのつもりで読んでみてもらいたい。各ルーチンごとにポイントとなる部分を拾って軽く解説しておく。

●エントリ～終了（62行～）

Interruptスイッチなどによってプログラムの実行が中断された場合に後始末をせずに親プロセスに帰るのがいやだったので、67～72行で前回のASX.Xとまったく同じ手順で中断時の戻りアドレスを77行のラベルbreakの位置に設定している。

break以降では諸々の後始末をするサブルーチンを呼び出してから、キーバッファをクリアし、exitで実行終了する。マウスしか使わないプログラムでキーバッファを気にしているのが変に見えるかもしれないが、“マウスしか使わないからこそ”この処理が必要なのだ。これを怠ると、プログラム走行中に誤って押されたキーがプログラム終了後にまとめて吐き出されることになる。

●初期化ルーチン（275行～）

278～290行でDOSコールconctrlによって画面モードを横512×縦512ドット、256色モードに切り換えたうえで、邪魔なファンクションキー行とカーソルを消している。画面モードとファンクションキー行についてはあとで元に戻せるように（374行以下の後始末ルーチン参照）現在の状況をワークエリアにしまっておく。それが作法というものだ。あと、このサブルーチンでは頭でlinkし、リターンする直前でunlinkすることによってDOSコール呼び出し時のスタック補正を省略するという姑息なテクニックが使われている。あまり褒められたことではないが、一度やって見せたかった。

293、294行は下位のサブルーチンを呼び出して、ペンパターンとして利用する外字の定義を行っている。ここでも、あとで元に戻せるように現在の外字の定義を取得・待避しておくのを忘れない。定義する外字のフォントパターンは436行以下に用意しており、16ワードが1文字分のデータにあたる。

頭に縦横のドット数がつけてあるのはほかとの兼ね合いで、実際には使っていない。フォントパターンは438～453行の最初の1個だけは見やすく2進数で表記してみた（2個目以降はスペースの都合で詰めて16進数で表記してある）。これを見ればフォントパターンの形式・作り方は一目瞭然だろう。

●メニューウィンドウの初期化（309行～）

前述のとおり、メニューはあらかじめ全部描いておく。描画に必要なデータはデータセクションに用意しておき、これを次々にIOCSコールに渡している。

●メイン処理 (88行～)

多少冗長な作りになっているが、マウスのボタンの状態をチェックし、ボタンが押されていたらその位置に応じてそれなりの処理を行うというパターンの組み合わせであり、リスト2と基本的には大差ない。左ボタンが押された場合は、まずメニューウィンドウ上かどうかを調べ、ウィンドウ外（もしくはウィンドウが非表示状態）であれば197行に飛んでSYMBOLで現在設定されているペンパターン（に対応する外字1文字）を描く。ウィンドウ上だった場合は、マウスカーソル座標から、

- 1) ペン選択メニュー上
- 2) 色選択メニュー上
- 3) 終了メニュー上
- 4) いずれでもないウィンドウの外枠

を識別し、対応する処理を行う。1), 2) の場合はさらにメニュー上のどの部分かの判定が加わることになる。また、ウィンドウの外枠で左ボタンが押された場合はウィンドウをドラッグするようにしてみた。本来ならマウスの動きに連動してリアルタイムでウィンドウの位置を変更することもできたのだが、もっと単純に、ボタンが離された位置へいきなりウィンドウを移動するようになっている。ここは読者に手を入れてもらいたい部分のひとつだ。

左ボタンの処理に比べれば、207行以下の右ボタンによるメニューウィンドウのON/OFF切り換え処理はシンプルだ。現在メニューが表示中かどうかを覚えておくワークmenuflagを調べて (209行)、もしメニューがすでに表示中であれば212行以下でVPAGEによりメニューが描かれているページを

非表示にする。メニューが表示されていなければ221行以下で現在のマウスカーソルの位置にメニューを表示する。

なお、222行でmenuflagをセットするのに使っているst.bは、任意の1バイトをFF_Hにする命令だ（オペランドサイズはバイト固定）。正確にはstの一般形はsXX（sはSetの略）であり、XXの部分には条件分岐命令同様の条件が入る。sXXは命令実行の時点でこの条件が成り立っていればオペランドをFF_Hにし、条件が成り立っていなければ00_Hにする命令で、stはこの条件が“t (always True: 常に真)”になった形だ。条件が常に成り立つわけだから、オペランドを00_Hにすることはありえない。逆にsfという命令は条件が“f (always False: 常に偽)”であり、任意の1バイトを00_Hにするのに使える。趣味の問題だが、人によってはclr.bの代わりに使うこともある。

*

『プログラマーズマニュアル』をパラパラと眺めてみると、それ単体でプログラムとして成り立つようなIOCSコールがいくつか見つかると思う。例を挙げるなら、コール番号7F_HのONTIME（本体を立ち上げてからの時間を100分の1秒単位で返す）とか8E_HのBOOTINF（前面の電源スイッチにより起動されたのか、タイマにより起動されたのか、また、どのデバイスから起動されたのかといったブート情報を返す）なんかは、IOCSコールからの戻り値を表示するだけでもそれなりに役にたつ（ことがあるかもしれない）プログラムになる。この類のプログラムはあって困るものでもなし、暇を見つけて作っておくとよいだろう。

来月は、グラフィックをもう少し本格的に取り上げる予定でいる。

リスト4 LINE.S

```

1:      .include      doscall.mac
2:      .include      iocscall.mac
3:      .include      const.h
4:      *
5:      .xref      atoi      *外部参照
6:      *
7:      PARCNT      equ      6      *IOCSに渡すパラメータの個数
8:      *
9:      .text
10:     .even
11:     *
12:     ent:
13:     lea.l      .mysp(pc),sp      *spを初期化する
14:     *
15:     bsr      getpar      *パラメータを取得する
16:     *
17:     moveq.l      #-1,d1      *グラフィック画面は
18:     IOCS      _APAGE      * 初期化されているか?
19:     tst.b      d0
20:     bmi      error      *未初期化ならエラー終了
21:     *
22:     lea.l      giospar(pc),a1      *直線描画
23:     IOCS      _LINE      *
24:     tst.b      d0      *エラー?
25:     bmi      usage      *パラメータの値が変
26:     *
27:     DOS      _EXIT      *正常終了
28:     *
29:     *
30:     *
31:     *      PARCNT個の値をバッファにセットする
32:     getpar:
33:     tst.b      (a2)+      *空文字列なら
34:     beq      usage      * 使用方法を表示して終了
35:     *
36:     lea.l      giospar(pc),a1      *a1=パラメータ格納バッファ
37:     moveq.l      #PARCNT-1,d1      *d1=ループカウンタ
38:     getpr0:  move.l      a2,-(sp)      *文字列→数値変換
39:     bsr      atoi      *
40:     movea.l      (sp)+,a2      *a2=続く文字列
41:     bmi      usage      *うまく変換できなかった
42:     move.w      d0,(a1)+      *パラメータを格納

```

```

43:     dbra      d1,getpr0      *PARCNT回繰り返す
44:     *
45:     rts
46:     *
47:     *
48:     *      使用法の表示&エラー終了
49:     *
50:     usage:
51:     move.w      #STDERR,-(sp)      *標準エラー出力へ
52:     pea.l      usgmes(pc)      * メッセージを
53:     DOS      _FPUTS      * 出力する
54:     addq.w      #6,sp      *
55:     *
56:     error:  move.w      #1,-(sp)      *終了コード1を持って
57:     DOS      _EXIT2      * エラー終了
58:     *
59:     *
60:     *      データ&ワーク
61:     *
62:     .data
63:     .even
64:     *
65:     usgmes:  .dc.b      '機 能: グラフィック画面に直線を描きます'
66:     CR,LF
67:     .dc.b      '使用法: LINE X0 Y0 X1 Y1'
68:     .dc.b      ' パレットコード ラインスタイル'
69:     .dc.b      CR,LF,0
70:     *
71:     .bss
72:     .even
73:     *
74:     giospar:
75:     .ds.w      PARCNT      *パラメータバッファ
76:     *
77:     .stack
78:     .even
79:     *
80:     mystack:  .ds.l      256      *スタック領域
81:     *
82:     mysp:
83:     .end      ent

```


リスト5 ATOI.S

```

1:      .include      const.h
2:      *
3:      .xdef      atoi
4:      *
5:      CCF      macro      *Cビットを反転するマクロ
6:      eor.w      #00001,ccr      *
7:      *      XNZVC      *
8:      endm
9:      *
10:     TOUPPER      macro      dreg      *英小文字→大文字変換マクロ
11:     local      skip
12:     cmpi.b      #'a',dreg      *
13:     bcs      skip      *
14:     cmpi.b      #'z'+1,dreg      *
15:     bcc      skip      *
16:     subi.b      #'a'-'A',dreg      *
17:     skip:
18:     endm
19:      *
20:      .text
21:      .even
22:      *
23:      *atoi(str)
24:      *機能: 数値を表す文字列を16ビット符号付整数に変換する
25:      *戻り値: d0.w = 変換された値
26:      *      (sp).l = 続く文字列へのポインタ
27:      *      N = 1文字も変換できなかった場合に1
28:      *メモ: 文字列先頭に余分な空白を置くことを許す
29:      *      先頭に'+','-'の符号をつけてもよい
30:      *      ex) 123, +123, -123
31:      *      's','x','X'をつけるのと16進数とみなす
32:      *      ex) $12AB, -$12AB, x12AB, X12AB
33:      *
34:      str      =      8
35:      *
36:      atoi:
37:      link      a6,#0      *スタックフレーム生成
38:      movem.l      d1-d3/a0,-(sp)      * [レジスタ待避
39:      movea.l      str(a6),a0      * a0=文字列へのポインタ
40:      bra      atoi1
41:      *
42:      atoi0:      addq.w      #1,a0      *文字列先頭の空白を
43:      atoi1:      cmpi.b      #SPACE,(a0)      * 飛ばす
44:      beq      atoi0      *
45:      cmpi.b      #TAB,(a0)      *
46:      beq      atoi0      *
47:      *
48:      moveq.l      #1,d2      *d2=符号(+)
49:      cmpi.b      #'+',(a0)      * '+'が指定されたか?
50:      beq      atoi2      *
51:      cmpi.b      #'-',(a0)      * '-'が指定されたか?
52:      bne      atoi3      *
53:      moveq.l      #-1,d2      *d2=符号(-)
54:      atoi2:      addq.w      #1,a0      *符号の分ポインタを進める
55:      *
56:      atoi3:      moveq.l      #0,d0      *結果を返すd0をクリア
57:      moveq.l      #0,d1      *作業用のd1をクリア
58:      moveq.l      #-1,d3      *仮にエラーフラグを立てる
59:      *
60:      cmpi.b      #'s',(a0)      *16進の指定かどうか調べず
61:      beq      htoi      *
62:      cmpi.b      #'x',(a0)      *
63:      beq      htoi      *
64:      cmpi.b      #'X',(a0)      *
65:      beq      htoi      *

```

```

66:      *
67:      bra      atoi5      *10進文字列のとき
68:      *
69:      atoi4:      addq.w      #1,a0      *1文字取り出す
70:      atoi5:      move.b      (a0),d1      *16進数値か?
71:      bsr      isdigit      * 数字か?
72:      bcs      atoiq      * そうでなければ終了
73:      mulu.w      #10,d0      *10進1桁分左にシフト
74:      swap.w      d0      * 上位ワードが
75:      tst.w      d0      * 0でなければ
76:      bne      atoi6      * オーバーフローした
77:      swap.w      d0      *
78:      add.w      d1,d0      * 下位に1桁追加
79:      moveq.l      #0,d3      *エラーフラグをクリア
80:      bra      atoi4      *繰り返す
81:      *
82:      *16進文字列のとき
83:      htoi:      addq.w      #1,a0      *1文字取り出す
84:      move.b      (a0),d1      *16進数値か?
85:      bsr      isxdigit      * 16進数か?
86:      bcs      atoiq      * そうでなければ終了
87:      asl.l      #4,d0      *16進1桁分左にシフト
88:      swap.w      d0      * 上位ワードが
89:      tst.w      d0      * 0でなければ
90:      bne      atoi6      * オーバーフローした
91:      swap.w      d0      *
92:      add.w      d1,d0      * 下位に1桁追加
93:      moveq.l      #0,d3      *エラーフラグをクリア
94:      bra      htoi
95:      *
96:      atoi6:      moveq.l      #-1,d3      *オーバーフローが発生
97:      atoiq:      mulu.w      d2,d0      * 符号を掛けて最終結果を得る
98:      move.l      a0,str(a6)      * 続く文字列へのポインタを返す
99:      tst.w      d3      * エラーフラグをccrに反映する
100:      movem.l      (sp),d1-d3/a0      * レジスタ復帰
101:      unlk      a6      * スタックフレーム解放
102:      rts
103:      *
104:      *
105:      *      10進文字→数値変換 (d1.b)
106:      *      (エラー時はC=1)
107:      isdigit:      subi.b      #'0',d1
108:      bcs      isdgtq
109:      cmpi.b      #9-1,d1
110:      CCF
111:      isdgtq:      rts
112:      *
113:      *
114:      *
115:      *      16進文字→数値変換 (d1.b)
116:      *      (エラー時はC=1)
117:      isxdigit:      TOUPPER      d1
118:      subi.b      #'0',d1
119:      bcs      isxdgq
120:      cmpi.b      #9-1,d1
121:      CCF
122:      bcc      isxdgq
123:      subq.b      #'A'-'0',d0,d1
124:      bcs      isxdgq
125:      cmpi.b      #15+1,d1
126:      CCF
127:      isxdgq:      rts
128:      *
129:      *
130:      .end

```

リスト6 STAMP.S

```

1:      .include      doscall.mac
2:      .include      iocscall.mac
3:      *
4:      CFKEYMOD      equ      14      *CONCTRLモード番号
5:      CSCREEN      equ      16      *
6:      CCURON      equ      17      *
7:      CCUROFF      equ      18      *
8:      *
9:      HIDEKEY      equ      3      *ファンクションキー非表示
10:     DOS_GM3      equ      4      *画面モード512x512,256色
11:     *
12:     DISABLEKEY      equ      0      *ソフトウェアキーボード禁止
13:     ENABLEKEY      equ      -1      *ソフトウェアキーボード許可
14:     *
15:     WINH      equ      272      *メニューウィンドウ幅
16:     WINV      equ      104      *メニューウィンドウ高さ
17:     *
18:     USERPAGE      equ      1      *描画を行う画面
19:     BIT_USERPAGE      equ      %0010      *
20:     MENUPAGE      equ      0      *メニューを表示する画面
21:     BIT_MENUPAGE      equ      %0001      *
22:     *
23:     *メニュー表示
24:     SHOWMENU      equ      BIT_USERPAGE|BIT_MENUPAGE
25:     *メニュー非表示
26:     HIDEMENU      equ      BIT_USERPAGE
27:     *
28:     GAIJITOP      equ      $eb9f      *全角外字の先頭文字コード
29:     FONT16      equ      $0008      *8x16,16x16
30:     *
31:     PATMAX      equ      8      *ペンパターンの最大数
32:     *
33:     *
34:     *グラフィック関係IOCSデータ受け渡し領域の構造
35:     *
36:     .offset      0
37:     *
38:     X0:      .ds.w      1      *POINT      *FILL      *BOX
39:     Y0:      .ds.w      1      *      *      *
40:     RETCOL:      *      *      *

```

```

41:     X1:      .ds.w      1      *      *      *
42:     POINTBUFSIZ:      *      *      *
43:     Y1:      .ds.w      1      *      *      *
44:     COL:      .ds.w      1      *      *      *
45:     FILLBUFSIZ:      *      *      *
46:     LS:      .ds.w      1      *      *      *
47:     BOXBUFSIZ:      *      *      *
48:     *
49:     *
50:     *      フォント読み込み領域の構造
51:     *
52:     .offset      0
53:     *
54:     XLEN:      .ds.w      1
55:     YLEN:      .ds.w      1
56:     FPAT:      .ds.w      15      *15x15
57:     FNTBUFSIZ:      *
58:     *
59:     .text
60:     .even
61:     *
62:     ent:
63:     lea.l      %rsp(pc),sp      *spを初期化する
64:     *
65:     bsr      init      *画面などの初期化
66:     *
67:     pea.l      break(pc)      *中断時の戻りアドレスを設定
68:     move.w      #CTRLVC,-(sp)      *
69:     DOS      _INTVCS      *
70:     move.w      #ERRJVC,(sp)      *
71:     DOS      _INTVCS      *
72:     addq.l      #6,sp      *
73:     *
74:     bsr      setupmenu      *メニューウィンドウの初期化
75:     bsr      main      *メイン処理
76:     *
77:     break:      bsr      windup      *後始末
78:     *
79:     move.w      #-1,-(sp)      *キーバッファクリア
80:     DOS      _XFLUSH      *

```

```

81:      addq.l  #2,sp      *
82:
83:      DOS      _EXIT      *終了
84:
85: *
86: *      メイン処理
87: *
88: main:
89:      IOCS      _MS_GETTDT      *ボタンの状態を取得
90:      tst.b     d0      *右ボタンが押されている?
91:      bne      rdown
92:      tst.w     d0      *左ボタンが押されている?
93:      bpl      main
94: *
95: ldown:      *左ボタンが押された
96:      IOCS      _MS_CURGT      *マウスカーソル位置を取得
97:      move.w    d0,d1      *d1.w = y
98:      clr.w     d0
99:      swap.w    d0      *d0.w = x
100:      tst.b     menuflag      *ウィンドウは表示中か?
101:      beq      pset
102:
103:      move.w    winx(pc),d2      *d2.w = ウィンドウ表示位置x
104:      move.w    winy(pc),d3      *d3.w = ウィンドウ表示位置y
105:
106:      cmp.w     d2,d0      *ウィンドウ上かどうかチェック
107:      bcs      pset
108:      cmp.w     d3,d1      *
109:      bcs      pset
110:      addi.w    #WINH,d2      *
111:      addi.w    #WINV,d3      *
112:      cmp.w     d2,d0      *
113:      bcc      pset
114:      cmp.w     d3,d1      *
115:      bcc      pset
116:
117:      *ウィンドウ内でクリックされた
118:      sub.w     winx(pc),d0      *d0.w = ローカルx座標
119:      sub.w     winy(pc),d1      *d1.w = ローカルy座標
120:      move.w    d0,pntbuf+X0      *x,yそれぞれを待避しておく
121:      move.w    d1,pntbuf+Y0
122:
123:      subq.w    #R,d0      *
124:      bcs      drag      *ウィンドウの左余白
125:      subq.w    #R,d1      *
126:      bcs      drag      *ウィンドウの上余白
127:      cmp.w     #256,d0      *
128:      bcc      drag      *ウィンドウの右余白
129:      cmp.w     #16,d1      *
130:      bcc      ldown1
131:
132:      cmp.w     #224,d0      *上段のメニュー内
133:      bcs      ldown0
134: done:
135:      rts      *終了ボックス内
136:
137: ldown0:      subi.w    #32,d0      *メインループを抜ける
138:      bcs      drag      *ペンメニューより左
139:      divu.w    #24,d0
140:      swap.w    d0
141:      cmpi.w    #16,d0
142:      bcc      drag      *ペンパターンの隣間の余白
143:      swap.w    d0
144: *
145: selpen:      *ペンメニュー内
146:      move.w    d0,d2      *d2.w = ペン番号
147:      addi.w    #GAIJITOP,d0      *新パターンを設定
148:      move.w    d0,curpat
149: *
150:      moveq.l   #MENUPAGE,d1      *メニュー用ページに
151:      IOCS      _APAGE      *切り換える
152:
153:      lea.l     current(pc),a1      *以前の枠を消す
154:      move.w    #255,COL(a1)      *
155:      IOCS      _BOX
156:
157:      mulu.w    #24,d2      *新たに枠を描く
158:      addi.w    #38,d2      *
159:      move.w    d2,X0(a1)      *
160:      addi.w    #19,d2      *
161:      move.w    d2,X1(a1)      *
162:      move.w    #1,COL(a1)      *
163:      IOCS      _BOX
164:
165:      bsr      lwait      *左ボタンが離されるのを待つ
166:
167:      bra      ldown2      *描画用ページに戻す
168:
169: ldown1:      subi.w    #24,d1      *
170:      bcs      drag      *ペンメニューと
171:      * 色メニューの隣間の余白
172:      cmpi.w    #64,d1      *
173:      bcc      drag      *ウィンドウの下余白
174: *
175: selcol:      *色メニュー内
176:      moveq.l   #MENUPAGE,d1      *メニュー用ページに
177:      IOCS      _APAGE      *切り換える
178:
179:      lea.l     pntbuf(pc),a1      *マウスカーソル位置から
180:      IOCS      _POINT      *色を拾う
181:      move.w    RETCOL(a1),d0      *
182:      move.w    d0,curcol      *カレントカラーにセット
183:
184:      lea.l     coldat(pc),a1      *カレントカラーで
185:      move.w    d0,COL(a1)      *メニュー左上の枠を
186:      IOCS      _FILL      *塗り潰す
187:
188: ldown2:      moveq.l   #USERPAGE,d1      *描画用ページに戻す
189:      IOCS      _APAGE
190:
191:      bra      main      *メインループへ
192: *
193: drag:      *ウィンドウの外枠でクリックされた
194:      bsr      lwait      *左ボタンが離されるのを待つ
195:      bra      menuon      *ウィンドウを描き直す

```

```

196: *
197: pset:      *ウィンドウ外でクリックされた
198:      lea.l     setdat(pc),a1      *マウスカーソルの位置に
199:      subq.w    #8,d0      *パターンを描く
200:      move.w    d0,X0(a1)
201:      subq.w    #8,d1
202:      move.w    d1,Y0(a1)
203:      IOCS      _SYMBOL
204:
205:      bra      main      *メインループへ
206: *
207: rdown:      *右ボタンが押された
208:      move.l     #(WINH/2)<<16,ofst      *表示位置オフセット
209:      tst.b     menuflag      *メニューは表示中か?
210:      beq      menuon
211: *
212: menuoff:      *メニューウィンドウを消す
213:      clr.b     menuflag      *フラグを抜かせる
214:      moveq.l   #HIDEMENU,d1      *メニューページ非表示
215:      IOCS      _VPAGE
216:
217:      bsr      rwait      *右ボタンが離されるのを待つ
218:
219:      bra      main      *メインループへ
220: *
221: menuon:      *メニューウィンドウを出す
222:      st.b      menuflag      *フラグを立てる
223:      IOCS      _MS_CURGT      *マウスカーソル位置を取得
224:      move.w    d0,d1      *d1.w = y
225:      swap.w    d0      *d0.w = x
226:
227:      sub.w     ofst+X0(pc),d0      *ウィンドウが
228:      bcc      mon0      *画面からはみ出さないよう
229:      clr.w     d0      *調整する
230:      sub.w     ofst+Y0(pc),d1
231:      bcc      mon1
232:      clr.w     d1
233:      mon1:      cmp.w     #512-WINH,d0
234:      bcs      mon2
235:      move.w    #512-WINH,d0
236:      mon2:      cmp.w     #512-WINV,d1
237:      bcs      mon3
238:      move.w    #512-WINV,d1
239:
240:      mon3:      move.w    d0,winx      *表示位置を格納
241:      move.w    d1,winy
242:
243:      moveq.l   #0,d2      *ウィンドウを目的の位置へ
244:      sub.w     d0,d2      *移動する
245:      andi.w    #511,d2
246:      moveq.l   #0,d3
247:      sub.w     d1,d3
248:      andi.w    #511,d3
249:      moveq.l   #BIT_MENUPAGE,d1
250:      IOCS      _HOME
251:
252:      moveq.l   #SHOWMENU,d1      *メニューページ表示
253:      IOCS      _VPAGE
254:
255:      bsr      rwait      *右ボタンが離されるのを待つ
256:
257:      bra      main      *メインループへ
258:
259: *
260: rwait:      *右ボタンが離されるのを待つ
261:      IOCS      _MS_GETTDT
262:      tst.b     d0
263:      bne      rwait
264:      rts
265: *
266: lwait:      *左ボタンが離されるのを待つ
267:      IOCS      _MS_GETTDT
268:      tst.w     d0
269:      bmi      lwait
270:      rts
271:
272: *
273: *      初期化
274: *
275: init:
276:      link      a6,#0
277:
278:      move.w    #DOS_QX3,-(sp)      *画面
279:      move.w    #CSCREEN,-(sp)      *画面を512x512,256色に
280:      DOS      _CONCTRL      *初期化
281:      move.w    d0,scrnmsav      *現在の画面モードを待避
282:
283:      move.w    #HIDEFKY,-(sp)      *ファンクションキー行を
284:      move.w    #CFREYMOD,-(sp)      *非表示に設定
285:      DOS      _CONCTRL
286:      move.w    d0,fkeymsav      *現在のファンクションキー行
287:      * モードを待避
288:
289:      move.w    #CCUROFF,-(sp)      *カーソル非表示モード
290:      DOS      _CONCTRL
291:
292:      bsr      savfont      *外字
293:      bsr      deffont      *待避
294:      * 定義
295:
296:      *マウス
297:      IOCS      _MS_INIT
298:      IOCS      _MS_CURON
299:      moveq.l   #DISABLESKEY,d1      *マウス初期化
300:      IOCS      _SKEY_MCD      *マウスカーソル表示
301:      * ソフトウェアキーボード
302:      * 表示禁止
303:      unlk      a6
304:      rts
305: *
306: *      メニューの初期化
307: *      (あらかじめ全部描いておく)
308: *
309: setupmenu:
310:      moveq.l   #MENUPAGE,d1      *メニュー用ページに

```



```

311:      IOCS      _APAGE      * 切り換える
312:
313:      moveq.l   #HIDEMENU,d1 *メニュー用ページ非表示
314:      IOCS      _VPAGE
315:
316:      lea.l     fildat(pc),a1 *ウィンドウ枠を塗り潰す
317:      IOCS      _FILL
318:
319:      lea.l     boxes(pc),a1 *BOXを必要なだけ描く
320: boxlp: tst.w     {a1}
321:      bmi       boxed
322:      IOCS      _BOX
323:      lea.l     BOXBUFSIZ(a1),a1
324:      bra       boxlp
325: boxed:
326:      lea.l     mendat(pc),a1 *ペンパターンメニューを
327:      IOCS      _SYMBOL      * 描く
328:
329:      bsr       makecoltbl * カラーテーブル
330:
331:      moveq.l   #USERPAGE,d1 *描画用ページに切り換える
332:      IOCS      _APAGE
333:
334:      clr.b     menuflag      *フラグを覆かせる
335:
336:      rts
337:
338: *
339: *      256色の色テーブルを描く
340: *
341: makecoltbl:
342:      link      a6,#-FILLBUFSIZ
343:
344:      lea.l     -FILLBUFSIZ(a6),a1 *a1=FILL用パラメータ領域
345:
346:      moveq.l   #0,d1
347:      *d1=色
348:      move.w     #32,Y0(a1)
349:      move.w     #32+7,Y1(a1)
350:      *{(8,32)-(8+7,32+7)から
351:      moveq.l   #8-1,d6
352:      move.w     #8,X0(a1)
353:      move.w     #8+7,X1(a1)
354:      *縦に8個
355:      moveq.l   #32-1,d7
356:      move.w     d1,COL(a1)
357:      IOCS      _FILL
358:      *
359:      addq.w     #8,X0(a1)
360:      addq.w     #8,X1(a1)
361:      addq.w     #1,d1
362:      dbra      d7,clp1
363:      *
364:      addq.w     #8,Y0(a1)
365:      addq.w     #8,Y1(a1)
366:      dbra      d6,clp0
367:      *繰り返す
368:      unlk      a6
369:      rts
370:
371: *
372: *      後始末
373: *
374: windup:
375:      link      a6,#0
376:
377:      move.w     scrnmsav,-(sp) *画面モードを戻す
378:      move.w     #CSCREEN,-(sp)
379:      DOS        _CONCTRL
380:
381:      move.w     fkeymsav,-(sp) *ファンクションキー行の
382:      move.w     #CFKEYMOD,-(sp) * モードを戻す
383:      DOS        _CONCTRL
384:
385:      move.w     #CCURON,-(sp) *カーソル表示モード
386:      DOS        _CONCTRL
387:
388:      bsr       rstfont      *外字フォント復帰
389:
390:      IOCS      _MS_INIT
391:      moveq.l   #ENABLESKEY,d1 *マウス初期化
392:      IOCS      _SKEY_MOD      * ソフトウェアキーボード
393:      * 表示許可
394:      unlk      a6
395:      rts
396:
397: *
398: *      外字の先頭8文字のフォントパターンを待避する
399: *
400: savfont:
401:      lea.l     fontbuf(pc),a1
402:      move.l     #FONT16<<16|GAIJITOP,d1
403:      moveq.l   #PATMAX-1,d2
404:      IOCS      _FNTGET
405:      addq.w     #1,d1
406:      lea.l     FNTBUFSIZ(a1),a1
407:      dbra      d2,savlp
408:      rts
409:
410: *
411: *      外字の先頭8文字にフォントパターンを設定する
412: *
413: deffont:
414:      lea.l     fontdat+FPAT(pc),a1
415:      move.l     #FONT16<<16|GAIJITOP,d1
416:      moveq.l   #PATMAX-1,d2
417:      IOCS      _DEFCHR
418:      addq.w     #1,d1
419:      lea.l     FNTBUFSIZ(a1),a1
420:      dbra      d2,deflp
421:      rts
422:
423: *
424: *      savfontで待避したフォントパターンを復帰する

```

```

425: *
426: rstfont:
427:      lea.l     fontbuf+FPAT(pc),a1
428:      bra       defnt0
429:
430: *
431: *      データワーク
432: *
433:      .data
434:      .even
435: *
436: fontdat:
437:      .dc.w     16,16
438:      .dc.w     %0000000000000000
439:      .dc.w     %0000000000000000
440:      .dc.w     %0000000000000000
441:      .dc.w     %0011111101111111
442:      .dc.w     %0010000001010001
443:      .dc.w     %0001000001010010
444:      .dc.w     %0001002001011100
445:      .dc.w     %0000100000101000
446:      .dc.w     %0000100000010000
447:      .dc.w     %0000010000010000
448:      .dc.w     %0000010000001000
449:      .dc.w     %0000101000001000
450:      .dc.w     %0001101000000100
451:      .dc.w     %0010010100000100
452:      .dc.w     %0100010100000010
453:      .dc.w     %0111110111111110
454:
455:      .dc.w     16,16
456:      .dc.w     %0000,%0000,%0000,%0000,%0000,%0000,%0000,%0000
457:      .dc.w     %0000,%0000,%0000,%0000,%0000,%0000,%0000,%0000
458:
459:      .dc.w     16,16
460:      .dc.w     %0000,%0000,%0000,%0000,%0000,%0000,%0000,%0180
461:      .dc.w     %0180,%0000,%0000,%0000,%0000,%0000,%0000,%0000
462:
463:      .dc.w     16,16
464:      .dc.w     %0000,%0000,%0000,%0000,%0000,%0000,%0000,%01c0
465:      .dc.w     %01c0,%01c0,%0000,%0000,%0000,%0000,%0000,%0000
466:
467:      .dc.w     16,16
468:      .dc.w     %0000,%0000,%0000,%0000,%0000,%0000,%0000,%0140
469:      .dc.w     %0000,%0140,%0220,%0410,%0000,%0000,%0000,%0000
470:
471:      .dc.w     16,16
472:      .dc.w     %0000,%0000,%0000,%0000,%0000,%0000,%0000,%0300
473:      .dc.w     %0180,%00c0,%0050,%0030,%0000,%0000,%0000,%0000
474:
475:      .dc.w     16,16
476:      .dc.w     %0000,%03e0,%07f0,%0fff,%0fff,%0fff,%0fff,%0fff
477:      .dc.w     %7fff,%7fff,%7fff,%7fff,%3ffe,%3ffe,%0fff,%03e0
478:
479:      .dc.w     16,16
480:      .dc.w     %0000,%7fff,%7fff,%7fff,%7fff,%7fff,%7fff,%7fff
481:      .dc.w     %7fff,%7fff,%7fff,%7fff,%7fff,%7fff,%7fff,%7fff
482:
483:      .dc.w     0,0
484:      .dc.w     WINH-1,WINV-1
485:      .dc.w     255
486:
487:      .dc.w     2,2,WINH-2-1,WINV-2-1,1,%ffff
488:      .dc.w     6,6,33,25,1,%ffff
489:
490:      .dc.w     38,6,57,25,1,%ffff
491:      .dc.w     230,6,255,25,1,%ffff
492:      .dc.w     6,30,255,97,1,%ffff
493:      .dc.w     -1
494:
495:      .dc.w     40,8
496:      .dc.l     patstr
497:
498:      .dc.b     1,1
499:      .dc.w     1
500:      .dc.b     1,0
501:
502:      .dc.w     8,8,31,23,255
503:
504:      .dc.w     0,0
505:      .dc.l     curpat
506:
507:      .dc.w     255
508:      .dc.b     1,0
509:
510:      .dc.b     %eb,%9f,%20,%eb,%a0,%20
511:
512:      .dc.b     %eb,%a1,%20,%eb,%a2,%20
513:      .dc.b     %eb,%a3,%20,%eb,%a4,%20
514:      .dc.b     %eb,%a5,%20,%eb,%a6,%20
515:      .dc.b     '終了',0
516:
517: *
518:      .bas
519:      .even
520: *
521:      .ds.b     FNTBUFSIZ*8
522:      .ofst:
523:      .ds.b     POINTBUFSIZ
524:      .ds.w     1
525:      .ds.w     1
526:      .ds.w     1
527:      .ds.w     1
528:
529:      .ds.b     1
530:
531:      .stack
532:      .even
533: *
534:      .ds.l     1024
535:
536:      .end      ent

```

PASCALのデータ型を見る

Fujii Yoshimi/Fujiki Takeshi

藤井義巳/藤木健士

連載も3回目になりますが、読者の皆さんはもうPASCALのプログラムをいくつか書いてみられたことと思います。Cを知っている人なら「なあんだ、簡単じゃねーか」と思われたでしょう。Cを知らない人でもそれほど難しくありません。ただ、PASCALは型の厳しい言語なので、型についてよく知っておかないとしょっちゅうコンパイラから文句をいわれます。たとえばCなら整数変数に実数値を代入しても、勝手に変換してくれていたのに、PASCALではエラーになるといった具合です。そこで今回はその型について、少し詳しく説明することになります。

データ型

WirthはPASCALをプログラミングの教育に使いたいと考えました。彼は著書『アルゴリズム+データ構造=プログラム』で、プログラムを作る際のデータ構造の大切さを教えています。その彼が設計したPASCALが豊富なデータ型を備えていたのは当然のこと、さまざまなデータ構造を直接に記述することができます。PASCALのデータ型はおおまかに単純データ型と構造データ型、およびポインタ型に分類することができます。単純データ型はさらに、整数型、実数型、列挙型、論理型、文字型、部分範囲型に分かれます(図1)。また、実数型以外の単純データ型は順序型とも呼ばれて、共通の特徴を持っています。構造データ型には配列型、ファイル型、集合型、レコード型があります。PASCALの構造データ型はPackedという形容詞をつけると、多少の速度を犠牲にしても主記憶を食わないように、詰め込んだデータ型になります。このあたりは処理系によって対応がまちまちですが、後述の文字列型に関しては必ずPackedと書かなければならないことになっています。

PASCALがデータ構造の表現力に優れているのは、レコード型とポインタ型のおかげです。リスト構造、ツリー構造、キューなどの基本的なデータ構造をレコード型とポインタなしで表現することを想像してみてください。なにを隠そうFORTRANの世界では、21世紀を迎えようとする今日になっても、「データ構造はすべて配列で作る」なんて野蛮なことが行われているのです。信じられませんね。

前置きはこれくらいにして、それぞれの型について説明していきましょう。

PASCALはさまざまなデータを多彩な方法で取り扱うことができます。それは整数や実数などの数値、文字列といったものから集合やポインタにまで及びます。それではPASCAL言語におけるデータの扱い方をまとめて見てみましょう。

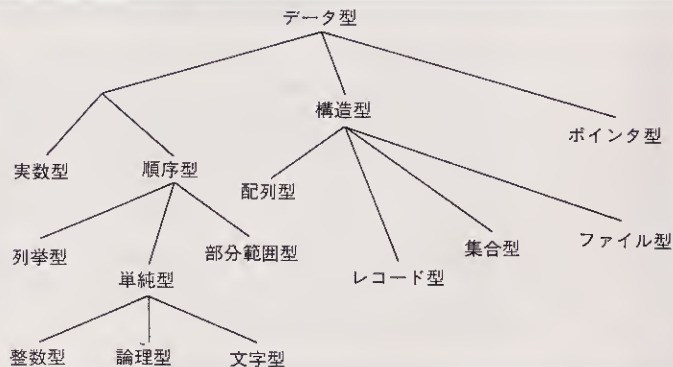
整数型と実数型

整数型と実数型はどの言語でもお馴染みの型ですね。PASCALの整数型の最大値は定義済み定数Max Intで知ることができます。例によって8086系CPUの処理系ではMax Intは32767であることが多く、PurePASCALは $2^{31}-1$ (計算して!)です。実数についてはなにもいうことはありません。式の中で実数と整数は混在して使うことが許されています。というより、整数は実数が必要とされる文脈では自動的に実数に変換されます。整数変数に実数値を代入することはできません。実数から整数に変換する方法はあとで説明します。整数型はInteger、実数型はRealという名前です。PurePASCALではReal型は32ビットの単精度のみ用意されています (PurePASCALは実数演算にFLOAT??Xを利用しています。FLOAT2+.XとFLOAT3+.Xは、それぞれFLOAT2.X, FLOAT3.Xよりも速いのですが、単精度浮動小数点演算にバグを持っており、PurePASCALでの実数演算でおかしい結果が出る場合があります)。

論理型

C言語やBASICでは論理式の値が整数になっていましたが、PASCALには独立した論理型が存在します。論理型は型名Booleanで定義され、TrueかFalseかどちらかの値を取ります。たとえばa=1, b=1のとき、式a=bはTrueで、a=-1, b=1のとき、式a>bはFalseです。

図1 PASCALのデータ型



文字型

文字型の変数には、

```
var c:char;
begin
    :
    c:='A';
    :
end
```

といったようにキャラクタコードが格納されます。Cの文字型は8ビットの整数として使われていましたが、PASCALでは独立した型で、整数との混用はできません。

文字列は後ろでも説明するつもりですが、文字型のPACKED配列として作ります。文字型の定数は'A'のように表現します。漢字などの2バイトコードは文字型には使えません。PurePASCALでは2バイトコードは、文字列の中でだけ使用可能です。

列挙型

列挙型はC言語でもお馴染みですね。知らない人のために少し説明します。プログラム中に定数値をマジックナンバーとして埋め込むと、わかりにくくなりがちです。

```
if data=-1 then
```

と書くよりも、

```
const ILLEGAL=-1;
```

と定数を定義しておいて、

```
if data=ILLEGAL then
```

と書いたほうがよいことはわかりますね。こうすると、あとで仕様変更してILLEGALの値が変わったときでも、最初の定数定義を変更するだけですみます。

この名前付きの定数と似た概念として、列挙型というものがあります。これは、

```
type DAYS=(Mon, Tue, Wed, Thu, Fri, Sat, Sun);
var date:DAYS;
```

のようにすることで、MonからSunまでの値を持つ新たなDAYS型を作ることができるものです。MonからSunまでのそれぞれは、順序数として、0から6までの整数値を持っています。順序数を取り出すには標準関数Ordを使います。名前付きの定数と列挙型の違いは、名前付きの定数というのは単に、定数値に別名をつけただけなのに対して、列挙型は既存のどの型とも違う新たな型を作り出します。そうすることでプログラムの安全性を高めているのです。論理型Booleanも次のように列挙型の一種と考えることができます。

```
type Boolean=(False, True);
```

よってOrd(False)=0、Ord(True)=1です。

部分範囲型

部分範囲型は、事前に定義された順序型のある範囲の値だけをとる型です。たとえば、

```
type Subrange=1..10;
    Weekday=Mon..Fri;
var t:Subrange;
    d:Weekday;
```

という具合に部分範囲型Subrange、Weekdayを定義すると、Subrange型の変数は1から10まで、Weekday型の変数dはMonからFriまでの値だけを代入することができるわけです。それ以外の値を代入したら実行時エラーになるでしょう。

このようなチェック機構もPASCAL系の言語の特徴です。バグのために変数に予期せぬ値が代入されるなんてよくある話です。こんなとき、Cだったらデバッグ用のprintfをたくさん入れて再コンパイルということを皆さんしているのではないのでしょうか。ソースコードデバッグがあれば少しはましでしょうが、それにしてもバグの箇所を特定するのにかなり苦労するでしょう。PASCALだったら多くの場合、ランタイムエラーで一発で見つかるわけです。C言語もこういった機構を取り入れて、

```
int i:1..10;
```

なんて記述ができればいいと思いませんか。チェックはコンパイルスイッチでいつでもoffできるわけですからね。それにいま、shortとかlongとか、ユーザーが決定しているのですが、このように書いたらコンパイラが勝手にintのサイズを決めてくれるわけです。いまCコンパイラを作っている人がいたら、ぜひ考えてください。規格のあとでいじりゃ面白くないでしょ。なんて書きましたが、PurePASCALもそのあたりはさぼっていて、部分範囲型も必ず2バイトを占めます。

配列型

配列型の定義は、

```
type 配列型名=array [添字型] of要素型;
というかならで行われます。要素型は任意の型、添字型は任意の順序型が指定できます。いくつかの例を挙げると次のようになります。
```

```
type arrayA=array [-1..10] of Real;
    arrayB=array [Boolean] of array [Char]
    of Integer;
    arrayC=array [1..10, 1..10] of Real;
    arrayD=array [DAYS] of Integer;
```

arrayAは添字の下限が-1、上限が10、要素が12の実数配列です。arrayBは2×256要素、arrayCは10×10要素の2次元配列です。

arrayCの表記は、

```
type arrayC=array [1..10] of array [1..10]
of Real;
```

のかたちの省略形です。つまり、多次元配列は「配列の配列」と解釈されます。

```
var A:arrayA;
    B1,B2:arrayB;
    C1,C2:arrayC;
    i:Integer;
    date:DAYS;
    E:array [-1..10] of Real;
```

のとき、

```
A [i] :=1.23;
B1 [False] :=B2 [True] ;
C1:=C2;
for date:=Mon to Sun do
    arrayD [date] :=0;
```

といったような操作が可能です。Cとは違って配列の代入が可能で、配列のすべての要素がコピーされます。また、iの値が-1..10のあいだにないときは実行時エラーとなります。

ここでひとつ注意しなければならない点があります。変数Aと変数Eは一見すると同じ型のように見えるのですが、

```
E:=A;
```

のような操作は許されません。実はAとEは別々の型なのです。もっと極端な例を示すと、

```
var va:array [1..10] of Integer;
    vb:array [1..10] of Integer;
```

このとき、vaとvbは別々の型になってしまうのです。もしvaとvbを同じ型にしたいなら、

```
var va, vb:array [1..10] of Integer;
```

または、

```
type array1:array [1..10] of Integer;
var va:array1;
    vb:array1;
```

と書いてください。このように、PASCALの型が同一かどうかの判断は、その構造で判断するのではなく、型の名前かまたは変数が宣言された場所で決まるのです。「面倒でも型には名前をつけろ」ということなのでしょう。これは配列に限ったことではなく、ほかの構造型でも同じです。気をつけましょう。

文字列型

PASCALには文字列のための特別な型は存在しません。Cと同じように、文字列は文字型の配列として表されます。もう少し厳密な定義を示します。

```
type StringN=Packed array [1..N] of Char;
var str:StringN;
```

ただし、 $1 < N$ という約束です。'Packed'を忘れてはいけません。このように定められた文字列型に関しては、同じ文字列型同士の代入、関係演算(inを除く)が許されています。この「同じ文字列型」というのは、簡単にいえばNが同じということです。長さが違う文字列型同士の演算、代入はできません。あまり融通がききませんね。文字列定数は(single quote)で囲んで表現します。

```
Const N=25;
type StringN=Packed array [1..N] of Char;
var str1, str2:StringN;
```

```
begin
    str1:='This is character string.';
    str2:='This is string too.'
end;
```

このように、文字列定数の長さが短いときは、帳尻合わせにスペースを入れてください。

集合型

集合型はPASCAL独特のデータ型です。順序型の値の集合をビット列で表現し、集合演算を行うことができます。ただし、PurePASCALではかなりの制限つきで、集合の要素は順序数が0~127のものしか許されません。つまり、集合変数が128ビット=16バイトで表現されるわけです。私自身はあまり利用しないのですが、集合演算が直接行えるのは便利なのかもしれません。集合型の定数は[]の中に、要素を,(カンマ)で区切って並べます。また、要素が連続している場合は途中の要素を全部書く代わりに[3, 10..20, 40]といった記述もできます。

例：

```
type SetOfDays=Set of DAYS;
var weekday, allday:SetOfDays;
begin
    allday:= [Mon..Sun] ;
    weekday:=allday- [Sat,Sun]
end;
```

レコード型

C言語の構造体に当たるもので、いくつかの変数をまとめてひとつの変数として扱うデータ型です。このレコード型と、次に説明するポインタ型を組み合わせると、PASCALは実にさまざまなデータ構造を表現できます。レコード型の定義は、

```
type レコード型名=record固定部 可変部 end;
```

のようになされます。可変部とは、C言語でいう共用体にあたるものです。ありふれた例ですが、複素数型の作り方を例にレコード型の説明をしましょう。


```

type Complex=record
    Re, Im;Real
end;
var a,b,c:Complex;
begin
    c.Re:=a.Re*b.Re-a.Im*b.Im;
    c.Im:=a.Re*b.Im+a.Im*b.Re
end;

```

レコード型Complexは2つのフィールドRe, Imを持っています。ReとImはそれぞれReal型です。Complex型の変数aのフィールドReをアクセスするにはa.Reのように、変数名のあとに.とフィールド名Reを書きます。また、レコード型の変数同士の代入もできます。レコード型の構文規則はかなり複雑なので、構文図をつけておきますから参考にしてください(図2)。

ポインタ型

ポインタ型の変数は対象型と呼ばれる型の変数のアドレスを保持します。C言語のポインタと違って、アドレスの値を整数値に変換したり、あるいはポインタ同士、ポインタと整数のあいだで演算したりすることはできません。また、配列とも関係ありません。ポインタに対して許される演算は、同じポインタ同士の一致と不一致、それからポインタがなにも指していないことを意味する定数Nilとの比較だけです。ポインタ型の定義は、

```

type DataPtr= ^Data;
Data=Record
    item:Real;
    next:DataPtr
end;

```

のように、対象型の前に^をつけます。また、対象型にアクセスするときは、ポインタ変数の後ろに^をつけて行います。標準手続きNew(p)は対象型の変数の領域を主記憶上に確保し、そのアドレスをポインタ変数pに格

納します。不要になった領域は標準手続きDispose(p)で解放します。図3に双方向リスト、二進木を図示します。これらのデータ構造は次のデータ型Tree, Listで表現できます(よく見たらTreeもListも型の構造は同じですね)。

```

type TreePtr= ^Tree
Tree=Record
    data:Item;
    left, right:TreePtr
end;
ListPtr= ^List
List=Record
    data:Item;
    prev,next:ListPtr
end;

```

ファイル型

ファイル型というのは文字どおりファイルを使うために用意されたデータ型です。ただ、PASCALのファイルの概念は現在のUNIXやMS-DOSのそれとは大きく隔たりがあって、そのままではあまり実用的ではないと思います。詳しくは参考文献を読んでいただくことにして、ここでは概念を示すにとどめます。

ファイルは次のように宣言されます。

```
var DataFile:File of Data;
```

PASCALのファイルはシーケンシャルファイルで、ファイル処理はいくつかの標準手続きによって行われます。標準手続きを簡単に説明すると、次のようになります。

Reset(f)	ファイルfを読み込みのために初期化する。
Rewrite(f)	ファイルfから要素をひとつ取り、その値をf^1に入れる。
Get(f)	ファイルfから要素をひとつ取り、その値を変数f^1に入れる。

図2 レコード型の構文

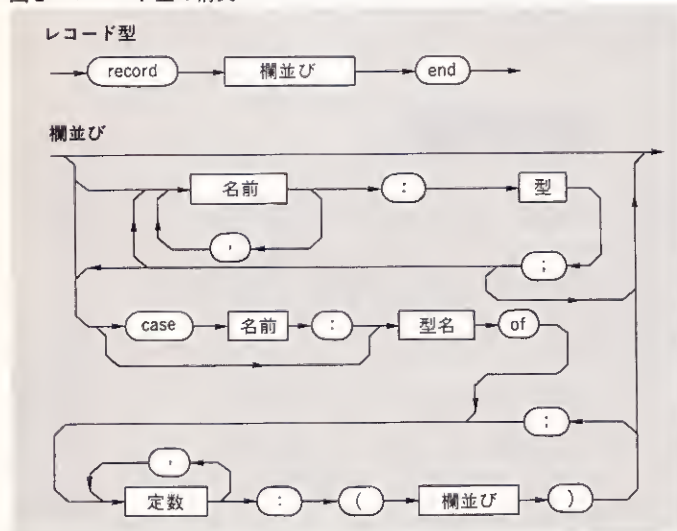
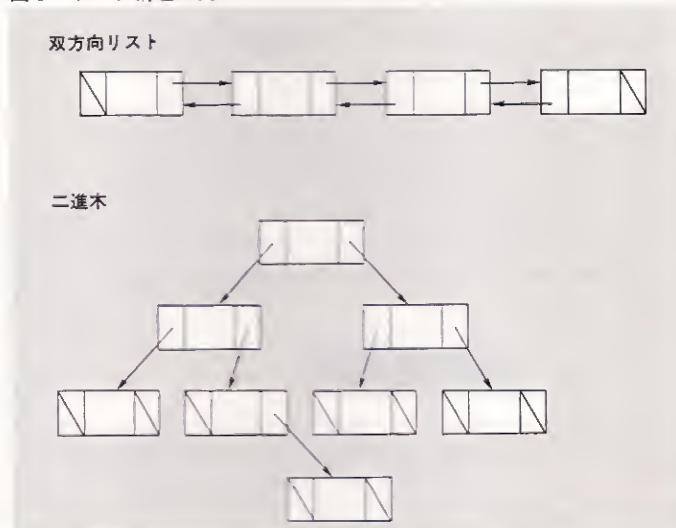


図3 データ構造の例



Put(f) バッファ変数 f^{\wedge} の値をファイルfに書き込む。

Read(f,x) ファイルfから要素をひとつ取り、その値を変数xに入れる。

Write(f,x) xの値をファイルfに書き込む。

fがファイル変数のとき、 f^{\wedge} (ポインタみたい) はバッファ変数と呼ばれ、ファイルを読み書きする場合、要素ひとつだけのバッファとなります。また、手続きRead, Writeは例外的に不定個の引数を取り、複数の要素を一度に読み書きできます。

```
write(f1,x1,x2,x3);
read(f2,y1,y2,y3);
```

は、

```
write(f1,x1);write(f1,x2);write(f1,x3);
read(f2,y1);read(f2,y2);read(f2,y3);
```

と同じことです。

プログラムの最初に、

```
program main(input, output);
```

と書きますが、このinputとoutputもファイル型の変数です。readとwriteの最初の引数を省略すると、readはinput, writeはoutputが指定されたと解釈されます。また、inputとoutputはテキストファイルと呼ばれる特殊なファイルで、readとwriteはこのテキストファイルに関しては特別な振る舞いをします。詳しくは次の機会に譲ります。

標準関数

PASCALにはいくつかの標準関数が用意されています。PASCALの標準関数はほかの言語と比較すると非常に少ないですが、教育用としては十分でしょう (表1)。それよりも問題となるのは、PASCALの標準関数のいくつかは、PASCAL自身で作ることができないという点です。たとえばPredとSuccは任意の順序型の値を引数として取れることになっていますが、PASCALの言語仕様では引数の型を複数指定することは許されていません。このような汚点があることをWirthは素直に認めていて、Modula-2では改善したようです。

Chrは一種の変換関数なのですが、C言語で言うところのキャスト演算子と解釈することもできます。事実、MacintoshのTHINK PASCALでは、Cだったら、

(型名) x

と書くところを、

型名 (x)

のかたちで、xの型を“型名”で示される型に変換することができますようです。整数型から文字型への変換が、ChrじゃなくてCharだったら完璧だったのにね。

実数を引数に取る関数は、整数も引数にできます。なぜなら整数は実数に自動的に変換されるからです。逆に実数→整数の変換は明示的に行う必要があり、そのために2種類の関数 (RoundとTrunc) が用意されています

表1 PASCALの標準関数

関数	引数	戻り値	説明
Abs(x)	実数型または整数型	実数型または整数型(xと同じ)	xの絶対値
ArcTan(x)	実数型	実数型	xの正接
Chr(i)	整数型	文字型	整数型→文字型変換
Cos(x)	実数型	実数型	xの余弦
Eof(f)	ファイル型	論理型	End of File
Eoln(f)	ファイル型	論理型	End of Line
Exp(x)	実数型	実数型	xの指数関数
Ln(x)	実数型	実数型	xの自然対数
Odd(i)	整数型	論理型	iが奇数なら真
Ord(o)	順序型	整数型	順序型→整数型
Pred(o)	順序型	oと同じ順序型	ひとつ前の要素を得る
Round(x)	実数型	整数型	xを四捨五入
Sin(x)	実数型	実数型	xの正弦
Sqr(x)	実数型	実数型	xの自乗
Sqrt(x)	実数型	実数型	xの2乗根
Succ(o)	順序型	oと同じ順序型	次の要素を得る
Trunc(x)	実数型	整数型	xを切り捨て

Round(x)	$0 \leq x$ のとき、 $x+0.5$ 以下の最大の整数
	$x < 0$ のとき、 $x-0.5$ 以上の最小の整数
Trunc(x)	$0 \leq x$ のとき、 x 以下の最大の整数
	$x < 0$ のとき、 x 以上の最小の整数

表2 RoundとTrunc

(表2)。

区別が面倒くさいかもしれませんが、 $0 \leq x$ のときは、Roundが四捨五入(丸め)、Truncが切り捨てと覚えておけば十分でしょう。

Ord(o)は順序型の順序数を調べる関数です。oが整数のときは当然 $o = \text{Ord}(o)$ が成り立ちます。ASCIIコードを採用している処理系ならば (PurePASCALはもちろん) $\text{Ord}('A') = 65$ ですね。

また、列挙型の場合、

```
type Colors=(Red, Blue, Green);
```

となっていたら、 $\text{Ord}(\text{Red}) = 0$, $\text{Ord}(\text{Blue}) = 1$, $\text{Ord}(\text{Green}) = 2$ となるでしょう。PASCALにはこのOrdの逆関数がないのも困りものです。このとき $\text{Pred}(\text{Blue}) = \text{Red}$, $\text{Succ}(\text{Blue}) = \text{Green}$ ということになります。Pred(Red)やSucc(Blue)はエラーです。論理型は、

```
type Boolean= (False, True);
```

と考えられるので、 $\text{Ord}(\text{False}) = 0$, $\text{Ord}(\text{True}) = 1$ となります。

* * *

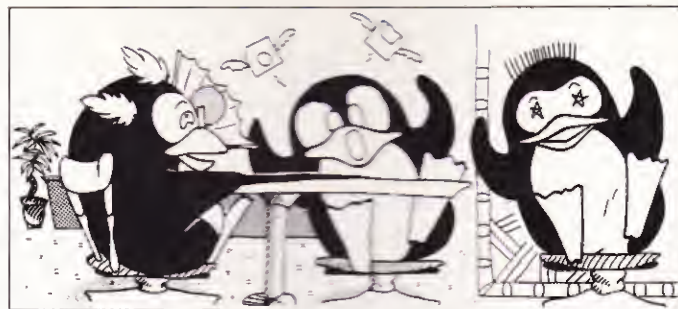
今月はPASCALの持つ豊富なデータ型について説明しました。本当は演算子の説明までやりたかったのですが、型の説明だけでかなり長くなってしまいました。演算子については次の機会に説明することにしましょう。

参考文献

- [1] Niklaus Wirth: "Algorithms + Data Structures = Programs", Prentice-Hall, 1976 (邦訳) 片山卓也: 「アルゴリズム+データ構造=プログラム」, 日本コンピュータ協会, 1979
- [2] Per Brinch Hansen: "Programming a Personal Computer", Prentice-Hall, 1983 (邦訳) 玄光男: 「パソコンシステムプログラム設計 I. コンパイラ設計編」, 電気書院, 1988

マシン語カクテル in Z80's Bar

第14回—— 楽な逆ポーランド？ ——



シナリオ&イラスト：山田純二

特別監修：浦川博之 金子俊一

カラコローン！

マスター（以下M）：いらっしやいませ。

純二（以下純）：どうぼ、ごんぱんばあー（どうも、こんばんは）。

ようこ（以下Yo）：あら、純二君おひさしぶり。どうしたの？ やばせばびのものまね？

長老（以下老）：それにしてもへたくそじやのう。もっと修業を積まんとお笑い芸人にはなれんぞ。

純：ぶたりども、びどいいがたでずね。ぼぐだって、ずぎでやっでるんじゃないんでず（2人とも、ひどい言い方ですね。僕だって、好きでやっでるんじゃないんです）。

善司（以下善）：趣味でやっでんでしょ。

老：同じじゃ、ばかもの。

純：がぜをびいでしまったんでずよ（かぜをひいてしまったんですよ）。

M：どうせ徹夜でポピュラスでもやっでいたんでしょ。

純：おおばたび！ どいだいですが、じつぱしがんごんばで、よどおじざわぎまぐって、あざおぎでみだらこうなっでいだんず。（大当たり！）と言いたいですが、実は新歓コンパで夜通し騒ぎまくって、朝起きたらこうなっでいたんです）。

老：理由はわかったが、そのしゃべり方はなんとかならんか。

純：ぜりぶがにばいばじになっでげんごうがずずんでいい、どらいだーばおもっでいぬようでずが（セリフが2倍増しになって原稿が進んでいい、とライターは思っているようです）。

M：そんなことよりツケの残りを早く払ってくださいよ。

純：えっ、なにいてんでず。先月はプログラムをちゃんと渡したじゃないですか（いきなり元の声に戻った）。

Yo：残念。前回の騒ぎ、全部純二君もち

になっているわよ。

純：そんな、バナナン、ばななん、ば、な。な。

善：空にキラキラお星様……。

Yo：あなたは寝ていなさい。

善：ぐうぐう……。

老：増える増える飲み屋のツケ。大きくなれよ。

純：大きくなっでたまりますか。長老が女の子の分はもって、あとはワリカンという話はどうなっちやっでたんでずか？

老：さーて、なんのことかな。わしゃ知らんぞ。

純：まったく、これだから年寄り様なんだよな。

老：ほっほほ。いくら反論してもツケは消えん、かんねんせい。

純：とほほ……。



電卓プロジェクト始動

老：ま、ツケの話はともかく、前回の騒ぎをやってもらおうかの。

M：電卓のプログラムですよね。

純：そうです。ちょっとリストが大きかったんで、今月まで残ちやっでたんでず。

老：数式をちゃんと記述できる電卓を作っておったんじやったな。その式の計算方法はどやっでおるのじゃ。

純：それはでずね。式をいったん逆ポーランド記法に変換してから計算して答えを求めています。

Yo：ふーん。ポーランド人が逆立ちでもするの？

純：いや、その場合ドンラーボ記法といったほうが……。

老：……逆ポーランド記法というのはじゃな。演算子には優先順位があるということを考えてできた計算法のことじゃま。

Yo：優先順位って掛け算と割り算は足し引きより先にやるっていうアレのこと？

純：そう。順次処理の好きなコンピュータには逆ポーランド記法が便利なんですよ。

Yo：へえ。なんで？

老：たとえば、 $10 + 5 \times 3$ という式は、まず 5×3 を計算してから10を足すじゃろう。こんなふうに式の中を先へ進んだり、元に戻ったりするとプログラムがややくしくなっでしまふのじゃ。

純：それが逆ポーランド表記ならさっきの式を例にやってみると、

$10 \ 5 \ 3 \ * \ +$
と表せるんですよね。

Yo：ただ順番を並べ替えただけじゃないの。

老：これにはちゃんとした意味があるのじゃま。純二君や、これを実際に計算することができまふの。

純：そうなん。計算にはスタックを使うんです。右から式の内容を調べて定数はスタックに積み直していき、演算子が見つかったらスタックから定数2つを取り出し計算します。その答えをまたスタックに積み直してこれを繰り返して繰り返せば出来上がり。

老：正確じゃ。このように逆ポーランド記法を使えば、式の中をいったりきたりせずに右から順番に処理することができまふのじゃま。わかったかな、ようこちゃん。

Yo：なるほど、便利にできてるのね。



どうすりゃいいんだ

M：では、逆ポーランド記法への変換のアルゴリズムはどうなっでいるんですか。

純：まず、式変換ワーク、演算ワークと、このプログラムの心臓部といえる変換テーブル（リスト、の682~694行参照）を用意し

ます。演算子をどういふ順番で書くかはこの変換テーブルが決めてくれるんです。

で、左から順番に式の項を読んでいき、定数は式変換ワークに出力。演算子だったらいま取り出した演算子を横の値、演算ワークのいちばん新しい演算子を縦の値として変換テーブルの内容を取り出します。この内容に従って次のような処理をしています。

00 演算ワークのいちばん新しいものを取り出して式変換ワークに出力。取り出した演算子はそのままで同じ処理を繰り返す。

01 取り出した演算子を演算ワークに出力。

02 演算ワークのいちばん新しいものを捨てて、新たに“*”を演算ワークに出力する。

03 演算ワークのいちばん新しいものを、ただ捨てる。

04 演算ワークのいちばん新しいものを捨てて、新たに“*”を2つ演算ワークに出力する。

05 終了

99 エラー

と、どんどん処理を繰り返していくと、式変換ワークに逆ポーランド記法の式が出来上がるわけです。

Yo: そんなふうにいわれても……。

純: わかりました。それじゃあ、

$10+20+30*40-50$

という式を変換していく様子を見ていきましょう。まず、10は式変換ワークに出力。次は演算子の“+”がくるので変換テーブルの内容を取り出しにいきます。テーブルの縦の値は“+”で、横の値は演算ワークのいちばん新しいもの（この場合はなにも入っていないからワークのエンドコード）。処理内容は01だから、“+”をそのまま演算ワークに出力します。

次の項は20。定数項だから式変換ワークに出力。2番目の演算子“+”のときも1番目と同様にしてテーブルの中身を見る。縦も横も“+”の場合、処理内容は00なので演算ワークのいちばん上の演算子を式変換ワークに出力する。この場合は式から取り出してきた演算子はそのままで、もう一度変換テーブルを見る。今度は縦が“+”，横がワークのエンドコードで処理内容は01となるので演算ワークに“+”を出力する。ここまでで式変換ワークと演算ワークにどのような出力がされているかわかりますか、ようこそさん。

Yo: 式変換ワークには10と20に最初の“+”があつて、演算ワークには2番目の

“+”があると思うけど。

純: 正解ですよ、ようこちゃん。これでもいいのの流れはつかめたでしょう。

老: それで最後には、

$10\ 20\ +\ 30\ 40\ *\ +\ 50\ -$
という式に変換されるわけじゃな。

Yo: なるほどね。



カッコがつくぞ～

M: 純二君、いままでの話を聞いているとわざわざテーブルとか用意なくてもプログラムで何とかかなりそうな気がするけど。

純: 確かにやっていることは優先順位に従って2つの処理を選択しているだけですからね。しかし、この変換テーブルのおかげで括弧を使った式の展開のプログラムがすっきり組めるんですよ。

老: まあ、テーブルを使わずにやってやれないことはないがリストは汚くなってしまいうじゃろな。

Yo: リストは読みやすいにこしたことはないってことね。

純: じゃ、話も一段落したところで、次はその括弧の話。括弧の使い方には3通りのパターンがあります。

1番目は、 $10+(20+30)$ のように、括弧の中の式をただ優先させるもの。

2番目は、 $10(20+30)$ または $(20+30)10$ のように、前か後ろのどちらかの括弧に乗算が省略されている場合。

3番目は、 $(10+20)(30+40)(50+60)$ のように、前後のカッコに乗算の省略が行われている場合。

プログラムはこのそれぞれの処理に対応させるために3つ用意します。

M: それを変換テーブルの処理番号02, 03, 04ですね。

純: そのとおり。それぞれ1番目には03, 2番目には02, 3番目には04が対応しています。

Yo: ただの括弧と乗算が省略された括弧はどうやって判別するの?

老: それは括弧がどこにでてくるかでわかるのじゃ。数式は定数と演算子が順番に並んでできておるじゃろう。式の解析のときに定数のところで現れたらそれはただの括弧、演算子があるべきところで現れる括弧は乗算が省略されている括弧というわけじゃな。

そして、閉じ括弧のほうは必ず演算子があるはずのところに現れるので、式のもうひとつ後ろの項を調べて判別しなければならぬ。「演算子、または閉じ括弧」とき

た場合はただの閉じ括弧で、「定数項、または開括弧」ときた場合は乗算が省略された閉じ括弧であると判別できるのじゃ。

Yo: 開括弧、閉じ括弧の両方について調べなくちゃならないのね。

純: 普通だったら乗算の省略は考える必要はないかもしれませんが、やっぱりいつも使っている式をそのまま記述できたほうが気持ちいいですからね。



電卓の使用法

純: じゃあ、最後に電卓の使用法を説明しましょう。

M: あれ、なんだか普通と逆のような気がしますねえ。

老: まあまあ、よいではないか。どちらかといえばアルゴリズムの解説がメインなんじゃから。

純: そういうこと。で、電卓で使えるコマンドは、

?……メモリの内容を表示

=……M1～5のメモリに、直前に計算した答えを代入

の2つです。計算はプロンプトに続いて数式を入力すれば、答えが10進と16進で表示されます。数式で使える定数は、10進定数、16進定数（頭に\$を付ける）とメモリのM1～5です。使える演算子は四則演算と余算の“MOD”になっていて、単項演算子や関数はサポートしていません。と、こんなところですよ。

老: 単項演算子や関数もサポートすれば完璧な電卓となったじゃろうに。肝心なところで手を抜きおって。

純: 関数をサポートすると式のネスティングまでやらなくてはなりませんから、それは勘弁してください。

M: というところで、今日はずいぶん頑張って説明しましたね。ごろうさんです。とりあえず前回の分のツケはこれで払ってもらいましょう。



純：わあーい。

老：じゃあ、そろそろ時間だし、わしは失礼させてもらうか。

純：あ、僕も帰ります。それじゃあ、また

いつか暇になったらやってきます。さようなら。

M：毎度どうも。……と。約1名おいていっっちゃったけど、どうしようか。

Yoこそ最後までいいんじゃない。明日はゴミの日だし。

善：ようどう……。

つづく

リスト1

```
9257 357
9257 358 :DENTAKU in Z80 Bar MAIN
9257 359 : 1990.5.1 by Junji
9257 360
9257 361 ORG SEBEND
9257 362
9257 363 ENTRY
9257 364 LD A,$0C
9259 CD F4 1F 365 CALL #PRINT
925C CD E2 1F 366 CALL #PRINT
925F 2A 2A 2A 20 367 DB *** DENTAKU in Z80 Bar *** , $0D
9263 44 45 4E 54
9267 41 4B 55 20
926B 69 6E 20 5A
926F 38 30 27 42
9273 61 72 20 2A
9277 2A 2A 0D
927A 00
927B 368
927B 369 MAIN2
927B 370 LD A,">"
927D CD F4 1F 371 CALL #PRINT
9280 11 8E 91 372 LD DE,LIGET
9283 CD D3 1F 373 CALL #GETL
9286 1A 374 LD A,(DE)
9287 FE 1B 375 CP $1B
9289 28 F0 376 JR Z,MAIN2
928B FE 3E 377 CP ">"
928D 20 EC 378 JR NZ,MAIN2
928F 13 379 INC DE
9290 1A 380 LD A,(DE)
9291 B7 381 OR A
9292 28 E7 382 JR Z,MAIN2
9294 FE 51 383 CP "Q"
9296 C8 384 RET Z
9297 21 7B 92 385 LD HL,MAIN2
929A E5 386 PUSH HL
929B ED 73 22 95 387 LD (ERR+1),SP
929F FE 3F 388 CP ">"
92A1 28 07 389 JR Z,WEMPRT
92A3 FE 3D 390 CP ">"
92A5 28 29 391 JR Z,WEMSET
92A7 18 73 392 JR @SHIKI
92A9 C9 393 RET
92AA 394
92AA 395 :MEMORY PRINT
92AA 396 WEMPRT
92AA 397 LD DE,WENDAT
92AD 06 05 398 LD B,05
92AF 0E 31 399 LD C,"1"
92B1 400
92B1 3E 4D 401 LD A,"M"
92B3 CD F4 1F 402 CALL #PRINT
92B6 79 403 LD A,C
92B7 CD F4 1F 404 CALL #PRINT
92BA 3E 3D 405 LD A,"-"
92BC CD F4 1F 406 CALL #PRINT
92BF 1A 407 LD A,(DE)
92C0 6F 408 LD L,A
92C1 13 409 INC DE
92C2 1A 410 LD A,(DE)
92C3 67 411 LD H,A
92C4 13 412 INC DE
92C5 D5 413 PUSH DE
92C6 C5 414 PUSH BC
92C7 CD 0C 94 415 CALL HXDECPRT
92CA D1 416 POP BC
92CB D1 417 POP DE
92CC 0C 418 INC C
92CD 10 E2 419 DJNZ WEM2
92CF C9 420 RET
92D0 421
92D0 422 :ANSWER TO MEMORY
92D0 423 MEMSET
92D0 424 INC DE
92D1 1A 425 LD A,(DE)
92D2 FE 4D 426 CP "M"
92D4 C2 12 95 427 JP NZ,ERROR
92D7 13 428 INC DE
92D8 CD D1 90 429 CALL NUM10
92DB DA 12 95 430 JP C,ERROR
92DE 3E 05 431 LD A,05
92E0 8D 432 CP L
92E1 DA 12 95 433 JP C,ERROR
92E4 7D 434 LD A,L
92E5 B7 435 OR A
92E6 CA 12 95 436 JP Z,ERROR
92E9 7C 437 LD A,H
92EA B7 438 OR A
92EB C2 12 95 439 JP NZ,ERROR
92EE 3E 4D 440 LD A,"M"
92F0 CD F4 1F 441 CALL #PRINT
92F3 7D 442 LD A,L
92F4 C6 30 443 ADD A,"0"
92F6 CD F4 1F 444 CALL #PRINT
92F9 3E 3D 445 LD A,"-"
92FB CD F4 1F 446 CALL #PRINT
92FE 7D 447 LD A,L
92FF 3D 448 DEC A
9300 CD 12 93 449 CALL WEMADR
9303 ED 5B 8C 91 450 LD DE,(ANSWER)
9307 73 451 LD (HL),E
```

```
9308 23 452
9309 72 453
930A EB 454
930B CD 0B 91 455
930E CD EE 1F 456
9311 C9 457
9312 458
9312 459
9312 21 7B 91 460
9315 461
9315 B7 462
9316 C8 463
9317 23 464
9318 23 465
9319 3D 466
931A 18 F9 467
931C 468
931C 469
931C 470
931C 21 06 92 471
931F 22 86 91 472
9322 21 42 92 473
9325 22 88 91 474
9328 CD 35 93 475
932B 21 06 92 476
932E 22 86 91 477
9331 CD DB 93 478
9334 C9 479
9335 480
9335 481
9335 1A 482
9336 FE 00 483
9338 20 04 484
933A 485
933A 06 0A 486
933C 18 47 487
933E 488
933E FE 29 489
9340 CA 12 95 490
9343 FE 28 491
9345 20 08 492
9347 13 493
9348 3E 06 494
934A CD 2E 00 495
934D 18 E6 496
934F 497
934F CD CA 54 498
9352 DA 12 95 499
9355 500
9355 1A 501
9356 B7 502
9357 28 E1 503
9359 FE 26 504
935B 20 1F 505
935D 13 506
935E 46 09 507
9360 1A 508
9361 FE 00 509
9363 28 20 510
9365 FE 20 511
9367 28 19 512
9369 FE 26 513
936B 20 19 514
936D 21 0D 90 515
9370 D5 516
9371 C5 517
9372 CD 5A 50 518
9375 C1 519
9376 D1 520
9377 34 0C 521
9379 522
9379 05 523
937A 18 05 524
937C 525
937C 21 0D 91 526
937F CD 5A 50 527
9382 3A 12 95 528
9385 529
9385 CD 50 94 530
9388 B7 531
9389 28 1D 532
938B 3D 533
938C 28 19 534
938E 3D 535
938F 28 1D 536
9391 3D 537
9392 28 1D 538
9394 3D 539
9395 28 1D 540
9397 3D 541
9398 28 1D 542
939A C3 12 95 543
939D 544
939D CD 31 90 545
93A0 C5 546
93A1 CD 03 90 547
93A4 C1 548
93A5 18 0E 549
93A7 550
```

:SHIKI NO END CODE

:SHIKI NO END CODE

:SHIKI NO END CODE

:)

:WORK NI SHUTGUYOXU

```

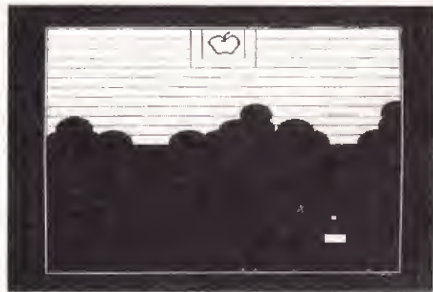
93A7 78          552      LD      A,B
93A8 CD 2E 90    553      CALL   ENZPUSH
93AB C3 35 93    554      JP      SHIKI
93AE          555      MULTSP
93AE CD 37 90    556      CALL   ENZPOP
93B1 3E 00       557      LD      A,$00
93B3 CD 2E 90    558      CALL   ENZPUSH
93B6 78          559      LD      A,B
93B7 FE 09       560      CP      $09
93B9 CA 55 93    561      JP      Z,SHIKI3
93BC C3 35 93    562      JP      SHIKI
93BF          563      SPDROP
93BF CD 37 90    564      CALL   ENZPOP
93C2 C3 55 93    565      JP      SHIKI3
93C5          566
93C5          567      DBMULTSP
93C5 CD 37 90    568      CALL   ENZPOP
93C8 97          569      SUB     A
93C9 CD 2E 90    570      CALL   ENZPUSH
93CC CD 2E 90    571      CALL   ENZPUSH
93CF C3 35 93    572      JP      SHIKI
93D2          573
93D2          574      ;SHIKI END
93D2          575      COMPLETE
93D2 3E FF       576      LD      A,$FF      ;WORK TO END CODE
93D4 01 FF FF    577      LD      BC,$FFFF
93D7 CD 40 90    578      CALL   WPRRITE
93DA C9          579      RET
93DB          580
93DB          581      ;KEISAN MAIN
93DB          582      CALSHIKI
93DB 21 06 92    583      LD      HL,CALSP
93DE 22 8A 91    584      LD      (CALADR),HL
93E1          585      CAL4
93E1 CD 4D 90    586      CALL   WPREAD
93E4 FE FF       587      CP      $FF
93E6 28 05       588      JR      Z,CAL2
93E8 CD 21 94    589      CALL   CALMAIN
93EB 18 F4       590      JR      CAL4
93ED          591      CAL2
93ED ED 73 FA 93 592      LD      (CAL3+1),SP
93F1 ED 7B 8A 91 593      LD      SP,(CALADR)
93F5 E1          594      POP     HL      ;ANSWER
93F6 22 8C 91    595      LD      (ANSWER),HL
93F9 31 00 00    596      LD      SP,$0000
93FC CD E2 1F    597      CALL   WMPRINT
93FF 41 4E 53 57 598      DB      "ANSWER- ",00
9403 45 52 3D 20
9407 00
9408 CD 0C 94    599      CALL   HXDECPRT
940B C9          600      RET
940C          601
940C          602      HXDECPRT
940C E5          603      PUSH   HL
940D CD 0B 91    604      CALL   STRING16
9410 3E 28       605      LD      A,"("
9412 CD F4 1F    606      CALL   #PRINT
9415 E1          607      POP     HL
9416 CD 0E 1F    608      CALL   #PRTHL
9419 CD E2 1F    609      CALL   WMPRINT
941C 48 29 0D 00 610      DB      "H)", $0D,00
9420 C9          611      RET
9421          612
9421          613      CALMAIN
9421 B7          614      OR      A
9422 28 07       615      JR      Z,PTEISU
9424 3D          616      DEC     A
9425 28 08       617      JR      Z,PMEMORY
9427 3D          618      DEC     A
9428 28 0E       619      JR      Z,ENZAN
942A C9          620      RET
942B          621      PTEISU
942B CD 0C 90    622      CALL   PUSHD
942E C9          623      RET
942F          624      PMEMORY
942F 79          625      LD      A,C
9430 CD 12 93    626      CALL   MEMADR
9433 4E          627      LD      C,(HL)
9434 23          628      INC     HL
9435 46          629      LD      B,(HL)
9436 18 F3       630      JR      PTEISU
9438          631      ENZAN
9438 21 52 94    632      LD      HL,EZ2
943B E5          633      PUSH   HL
943C 00          634      LD      H,B
943D 69          635      LD      L,C
943E 29          636      ADD    HL,HL
943F 11 71 91    637      LD      DE,JUMPTBL
9442 19          638      ADD    HL,DE
9443 4E          639      LD      C,(HL)
9444 23          640      INC     HL
9445 46          641      LD      B,(HL)
9446          642
9446 C5          643      PUSH   BC
9447 CD 1D 90    644      CALL   POPD
944A 59          645      LD      E,C
944B 50          646      LD      D,B
944C CD 1D 90    647      CALL   POPD
944F 69          648      LD      L,C
9450 60          649      LD      H,B
9451 C9          650      RET
9452          651      EZ2
9452 4D          652      LD      C,L
9453 44          653      LD      B,H
9454 CD 0C 90    654      CALL   PUSHD
9457 C9          655      RET
9458          656
9458          657      ;TABLE NUMBER GET
9458          658      TBLNUM
9458 D5          659      PUSH   DE
9459 21 72 94    660      LD      HL,ENZTBL

```

```

945C C5          661      PUSH   BC
945D 78          662      LD      A,B
945E          663
945E B7          664      OR      A
945F 11 08 00    665      LD      DE,08
9462 28 03       666      JR      Z,TBNUM2
9464          667      TBNUM3
9464 19          668      ADD    HL,DE
9465 10 FD       669      DJNZ   TBNUM3
9467          670      TBNUM2
9467 C1          671      POP     BC
9468 E5          672      PUSH   HL
9469          673
9469 2A 88 91     674      LD      HL,(ESPADR)
946C E2          675      LD      E,(HL)
946D E1          676      POP     HL
946E 19          677      ADD    HL,DE
946F 7E          678      LD      A,(HL)
9470 D1          679      POP     DE
9471 C9          680      RET
9472          681
9472          682      ENZTBL
9472 00 00 00 01 683      DB      * / MOD + - * ( ( SN
9476 01 01 01 01 684      DB      00,00,00,01,01,01,01,01 :*
947A 00 00 00 01 684      DB      00,00,00,01,01,01,01,01 :/
947E 01 01 01 01 685      DB      00,00,00,01,01,01,01,01 :MOD
9482 00 00 00 01 685      DB      00,00,00,00,00,01,01,01 :+
9486 01 01 01 01 686      DB      00,00,00,00,00,01,01,01 :-
948A 00 00 00 00 687      DB      01,01,01,01,01,01,01,01 :*(
948E 00 01 01 01 687      DB      01,01,01,01,01,01,01,01 : (
9492 00 00 00 00 687      DB      01,01,01,01,01,01,01,01 :DUMMY
9496 00 01 01 01 687      DB      00,00,00,00,00,04,02,99 :)*
949A          688
949A 01 01 01 01 689      DB      00,00,00,00,00,02,03,99 :)
949E 01 01 01 01 690      DB      00,00,00,00,00,99,99,05 :SPEND
94A6 01 01 01 01 691
94AA 01 01 01 01 691
94AE 01 01 01 01 692
94B2 00 00 00 00 693
94B6 00 94 02 53 694
94BA 00 00 00 00 695
94BE 00 02 03 53 695
94C2 00 00 00 00 695
94C6 00 53 53 05 695
94CA          696      ;TEISU NO HANTEI
94CA          697      TEISU
94CA CD 04 95    698      CALL   TE1CHK
94CD DA          699      RET     C
94CE 1A          700      LD      A,(DE)
94CF FE 4D       701      CP      "M"
94D1 28 0D       702      JR      Z,TEISU2
94D3 FE 24       703      CP      "$"
94D5 28 1B       704      JR      Z,NEXTE1
94D7 CD D1 90    705      CALL   NUM10
94DA D8          706      RET     C
94DB          707      DECTEI
94DB 97          708      SUB     A
94DC 4D          709      LD      C,L
94DD 44          710      LD      B,H
94DE 18 0E       711      JR      TEISU3
94E0          712      TEISU2
94E0 47          713      LD      B,A
94E1 13          714      INC     DE
94E2 1A          715      LD      A,(DE)
94E3 D6 31       716      SUB     "1"
94E5 D8          717      RET     C
94E6 FE 04       718      CP      04
94E8 3F          719      CCF
94E9 D8          720      RET     C
94EA 13          721      INC     DE
94EB 4F          722      LD      C,A
94EC 3E 01       723      LD      A,01
94EE          724      TEISU3
94EE CD 40 90    725      CALL   WPRRITE
94F1 C9          726      RET
94F2          727      HEXTEI
94F2 13          728      INC     DE
94F3 D5          729      PUSH   DE
94F4 CD B2 1F    730      CALL   #HLHEX
94F7 C1          731      POP     BC
94F8 30 E1       732      JR      NC,DECTEI
94FA 59          733      LD      E,C
94FB 50          734      LD      D,B
94FC CD B5 1F    735      CALL   #2HEX
94FF 26 00       736      LD      H,00
9501 6F          737      LD      L,A
9502 18 D7       738      JR      DECTEI
9504          739
9504          740      TEICHK
9504 1A          741      LD      A,(DE)
9505 FE 4D       742      CP      "M"
9507 C8          743      RET     Z
9508 FE 24       744      CP      "-s"
950A C8          745      RET     Z
950B FE 30       746      CP      "-0"
950D D8          747      RET     C
950E FE 3A       748      CP      "-9"+1
9510 3F          749      CCF
9511 C9          750      RET
9512          751
9512          752      ;ERROR PRINT
9512          753      ERROR
9512 CD EE 1F     754      CALL   #LTNL
9515 CD E2 1F     755      CALL   WMPRINT
9518 45 52 52 4F 756      DB      "ERROR!!", $0D
951C 52 21 21 0D 757
9520 00          757      DB      00
9521 31 00 00     758      LD      SP,$0000
9524 C9          759      RET

```

かべくずし

パドルを動かしてタマに当てて、画面上半分の壁をくずし、画面中央の絵を破壊する。そう、早い話がブロックくずしなんです。なーんだと思ったそこのあなた、あなたは甘い、甘すぎるっ！ つい力が入ってしまいましたが、そこはショートプロに載るプログラム、当然ただのブロックくずしではありません。

まず、パドルはジョイスティックで8方向に動かすことができます。そう、左右だけに動くのではないんです。んで、さらにジョイスティックのボタンを押しながらだと速く動いて、それそれしたタマと追いかけてこごだ、てなもんです(ちなみにボタンを2つとも押しちゃうと、ほとんど操作不能なくらい速くなる)。ほらねっ、普通のとはちょっとばかし違うでしょ。

うーん、それはともかく、タマが壁にぶつかったときがいいですねえ。普通みたいになたった1ブロックが消えるんじゃないくて「バババッ」と爆発して、まるーくえぐれるんですね。そして、ブロックくずしの様に玉がブロックとブロックの間で往復運動したりもする(この表現でわかるかな?)わけなんです、このときなんか爆発してるのがとーっても綺麗です。そうそう、ターボボタン(ジョイスティックのトリガーを押すとパドルの移動が速くなる)機能は私が勝手につけてしまいました。作者の太田さん、ごめんなさい。

ちなみにどうやったかという、パドルの移動ルーチン(360,370行)のところで、strig(1)関数で取ってきた値をパドルの移動の増分値にかけてるだけです(だから、ボタンの左右どちらが押されたかで速さがちがう。そのうえ、同時に押すと上記のようにめちゃ速くなるわけ)。

いやそれにしても、ゲーム自体もたいしたものだけど、それ以上によくプログラムを小さくまとめたなーと感心してしまいました。特に、破壊目標である絵。よくこの小さいプログラムでこんな絵を表示させたねー。ええ、この短いプログラムのどこにそんな絵のパターン(しかも毎回毎回

リスト1 THE FANFAN

```

10 CLS4:WIDTH 40:INIT:DIMA(7,7),B(7,7):PRW 254:HI=0:REPEAT OFF
20 LINE(24,16)-(192,183),PSET,B
30 LINE(80,16)-(136,183),PSET,B
40 LINE(24,72)-(192,127),PSET,B
50 FOR I=1 TO 7:COLOR 2:PRINT " " :NEXT
60 GET@(0,0)-(7,7),A:CLS
70 GET@(0,0)-(7,7),B
80 COLOR 4:PRINT "----- THE FAN FAN -----":COLOR 5
90 N=3:F=0:F1=1:LE=1:SC=0:OV=5
100 GOSUB 540
110 LOCATE 25,21:PRINT "HIT SPACE KEY"
120 IF INKEY$=" " THEN 130 ELSE 120
130 LOCATE 25,21:PRINT " "
140 FOR L=0 TO N
150 XX=INT(RND*3):YY=INT(RND*3)
160 IF XX=0 THEN X=3
170 IF XX=1 THEN X=10
180 IF XX=2 THEN X=17
190 IF YY=0 THEN Y=2
200 IF YY=1 THEN Y=9
210 IF YY=2 THEN Y=16
220 PUT@(X,Y)-(X+7,Y+7),A:PLAY "L0GFG"
230 PUT@(X,Y)-(X+7,Y+7),B
240 IF F=1 THEN RETURN
250 IF X=3 AND Y=2 THEN K=7
260 IF X=3 AND Y=9 THEN K=4
270 IF X=3 AND Y=16 THEN K=1
280 IF X=10 AND Y=2 THEN K=8
290 IF X=10 AND Y=9 THEN K=5
300 IF X=10 AND Y=16 THEN K=2
310 IF X=17 AND Y=2 THEN K=9
320 IF X=17 AND Y=9 THEN K=6
330 IF X=17 AND Y=16 THEN K=3
340 POKE &HC000+L,K
350 NEXT
360 FOR L=0 TO N
370 K$=INPUT$(1)
380 K1=PEEK(&HC000+L):F=1
390 IF K$="1" THEN X=3 :Y=16:K2=1
400 IF K$="2" THEN X=10:Y=16:K2=2
410 IF K$="3" THEN X=17:Y=16:K2=3
420 IF K$="4" THEN X=3 :Y=9 :K2=4
430 IF K$="5" THEN X=10:Y=9 :K2=5
440 IF K$="6" THEN X=17:Y=9 :K2=6
450 IF K$="7" THEN X=3 :Y=2 :K2=7
460 IF K$="8" THEN X=10:Y=2 :K2=8
470 IF K$="9" THEN X=17:Y=2 :K2=9
480 IF K2=K1 THEN SC=SC+5:GOSUB 220 ELSE PLAY "DCD":L=L-1:OV=OV-1
490 GOSUB 540
500 IF OV=0 THEN 600
510 NEXT:F=0:F1=F1+1
520 IF F1>2 THEN LE=LE+1:N=N+1:F1=1
530 GOTO 100
540 IF HI<SC THEN HI=SC
550 LOCATE 25,3:PRINT "LEVEL ";LE
560 LOCATE 25,6:PRINT "HI-SCORE":HI
570 LOCATE 25,9:PRINT "SCORE ";SC
580 LOCATE 25,18:PRINT "FANFAN ";OV
590 RETURN
600 LOCATE 25,21:PRINT "++GAME OVER++"
610 FOR T=0 TO 5000:NEXT:CLS:GOTO 80

```

リスト2 かべくずし

```

10 int a,b,ch,ta,k,x,y,m,n,v,w,i,j,sc,sa,hs,sg:dim char z(255)
20 color 3:screen 0,1,1,1:apage(1):box(0,0,255,255,1):apage(0)
30 sp_init(1):sp_disp(1):sp_on(0,1):m alloc(1,10)
40 for a=0 to 2:for b=0 to 2:z(a*16+b)=14-(b=1):next:next:sp_def(1,z)
50 for a=0 to 7:for b=0 to 14:z(a*16+b)=14-(a=2)-(a=3):next:next:sp_def(0,z)
60 randomize(543+atod(right$(time$*2))):ch=int(rnd()*191)+165
70 while 1
80 color 7:sc=0:ta=0
90 while 1
100 start():if game() then break
110 for a=0 to 9999:next
120 ta=ta+20:if ta>120 then sc=sc+50000:ta=0
130 endwhile
140 for a=0 to 127:box(a,a,255-a,255-a,0):next
150 locate 0,1
160 if sc=hs then hs=sc:print "こりゃあすごいなハイスコアだよ!" else {
170 print "めざすは";hs:"がんばろう!" }
180 color 3:repeat:until strig(1)
190 endwhile
200 end
210 func start()
220 sa=10:v=int(rnd()*221)+16:w=248:i=0:j=0:v=v-y:z=16:m=4*int(rnd()*2)-2:n=-3
230 ch=ch+1:if ch=184 then ch=165
240 wipe():cls:locate 0,0:print sc
250 a=int(rnd()*192)

```


絵が違う)があるんだ? と、思わず探してしまっただけじゃないですか。ふーん、そうか。X68000にはこういう絵のパターンがちゃんとあって、そうやると絵が出せるんですか。私は知りませんでした。ねえねえ、太田さん、どこでこんな方法を知ったのか。こっそり教えてくださいませんか? (←外字定義をしたことのないやつ)

ただ、このゲームよくできているんだけど、壁に当たったときパドルの動きが止まっちゃうのと、あとリスト中に全然注釈がないのがちょっと残念なんだよねー。これから投稿する人はぜひプログラムに注釈をつけてくださいな。

うーむ、それにしてもこのコーナーはいつまで続けられるのかなー。とりえずマシン語カクテルとはスタートがほとんど同じなので負けたくないな。そんなこんなでまた来月。

```
260 for b=2 to 15:palet(b,hsv(a,31,71):sa=a-w:sa=191:next
270 a=a+66:a=a+192*(a>191):palet(1,hsv(a,1,0))
280 for a=0 to 11:fill(2,a*10+2:sa,0:sa=191:next
290 box(104,1,151,41,0):fill(105,2,10,0)
300 box(112,1,143,31,0):fill(113,2,10,0)
310 symbol(116,3,chr$(235)+chr$(chr$(1,1,0,0,0)
320 sp_move(0,v-7,w,0):sp_move(1,x-7,v-1):repeat until stick(1)
330 endfunc
340 func game()
350 k=stick(1):sg=stg(1)
360 i=(3+sg*3)*((k=1)+(k=2)+(k=7)-(k=8)-(k=9)-(k=10)):v=i
370 if v and 256 then v=v-i
380 j=(k>6)-(k<4)+(k=0) shl (2+sg):w=v-1:if w and 128 then w=w-j
390 x=x+m:if x and 256 then m=-m:m=x+m
400 if point(x,y) then m=-m:if dokan(point(x,y)) then return(0)
410 y=y+n:if y and 256 then if y<0 then m=0:w=0:else return(1)
420 if point(x,y) then n=-n:if dokan(point(x,y)) then return(0)
430 sp_move(1,x-1,y-1,1):sp_move(0,v-7,w,0)
440 if abs(v-x)<9 and abs(w-y)<4 then pakon()
450 goto 350
460 endfunc
470 func dokan(a)
480 for b=1 to a:circle(x,y,b,1):next
490 m_init()
500 if a=15 then sa=sa+11111:m_trk(1,"V10@68020") else m_trk(1,"V10@68020")
510 m_play():sc=sc+sa:locate 0,0:print sc:sc=0:sa=sa+11
520 for b=1 to a:circle(x,y+1,b,0):circle(x,y,b,0):next
530 return(a=15)
540 endfunc
550 func pakon()
560 if n<0 then return(1)
570 m_init():m_trk(1,"V10@6706E"):m_play()
580 sa=10:n=-3:if abs(i+m)=1 then m=-m
590 endfunc
```

1周年特別企画——どんちゃん騒ぎの部屋

えー、ささやかながら1周年特別企画として(手前ミソくさくてちょっと恥ずかしいんだけど)皆様からショートプロばかりに寄せられたご意見、ご感想、文句に苦情、祝辞の言葉などなどにお答えしたいと思います。最初の方、どぞ!

☆ほう。古村氏の連載も1周年ですか。月日は百代の過客にして天上下唯我独尊。で、人間には3種類あると仮定しよう。(1)普段は無口だが、文章を書かせるととても面白くて含蓄深いことを書く奴、(2)普段のノリがそのまま文章に出る(それ以外は書けなかったりする)ヤツ、(3)普段は面白いのに、文章はまったくつまらないやつ、である。(で)君はといえば、いわゆるがな(2)である。彼はあのとおり的人格なのだから、というわけで、変にウケ狙いなどせず、すくすく伸びて、成長した姿を読ませてほしい。それが非常に楽しみである(爆笑)。(荻窪圭)

へへーっ! 荻窪師匠からの祝辞だー! いつもお世話になってます。そーです、私はそれ以外は書けないんです。ちなみに荻窪師匠は(4)書いている文章もすごいが本物に会ってみるとさらにすごいので恐れ入ってしまうタイプ。つまり人間がはるかに深い(人物から文章の想像はつくけど文章から人物を想像できない)っていうことで尊敬しています。はい。今度飲みにいきましょうよ、師匠。もちろん荻窪師匠のおごりでね! しかし、悪友金子俊一とかグラフィックの魔術師丹明彦さんにも祝辞を頼んだのになー。いったいどーなってるんだろ。

☆(で)さんの初登場は(ビー)年(ビー)月号の(ビー)のレビューではないですか? 違っていたらすいません。

(アンケートハガキより、原正人さん)
ピンポンピンポン! 大正解です。えー、あの頃は(で)って使ってなかったのによくわかりましたねー。まだ、Oh!X編集部がいまの泉岳寺に移るまえのまえ、四番町の半地下の編集部頃の話だからねー。懐かしいなあ。ちなみに

本文中の「ビー」は私がつけたものですが、別に恐ろしいことが書いてあるわけではありません。あしからず。

☆5月号の「空飛ぶDNAデモ」を走らせてみた。それを見た友人曰く、「まんが日本昔ばなし」のオープニングみたいだと。

(アンケートハガキより、神生直敏さん)
おー。「ぼうやー、よい子だ〜♪」というあれですね(そういえばパロディで「ぼうやーよい子だ金だしな〜」というのがあったな)。しかし、あのデモは本当に好評でした。そうそう、某MS-DOSマシンにも似たようなデモがあるという話を聞いたのである人に見せてもらったのですが、見た瞬間、「勝った!」と思ってしまいました。

☆ちょーどゲームでも作ってみようと思ってたところなんです、(で)さん。シューティングじゃないけど。4月号の外部関数は役に立ちそうです。(アンケートハガキより、小林到さん)
わーい、それはよかった。うれしいです。ばーいハNZはなかなか評判がよいので喜んでおります。それに4月号のsp_chk(も)5月号のデモに負けず劣らず好評でした。ここそこい投稿が多い。ゆえにショートプロの評判も上がるというわけでとてもうれしい。小林さんまぜひ投稿してみてくださいな。

☆ライターのプロフィールが知りたい。

(アンケートハガキより、桐山秀幸さん)
ショートプロとは関係ないけど、思わず持ってきてしまいました。あははは、私も知りたい。うちの編集部は謎の人物がいっぱいいるから辻絶なものになること間違いなし。でも、自分のプロフィールは遠慮したい……。

☆すいませんー。Reserved featureエラーがでるんですけどー。(バグ電話より)

すいませんー。X1のBASIC(CZ-8FB01)には新旧のBASIC(ver.1と2)があるのはご存じですよ。ショートプロのものはほとんどがどちらのBASICでも使えるのですが、5月号のDIG

MANはBASIC専用だったんですよー。うっかり私で書き落れてたんです。本当にごめんなさい。今後のために(やらないように心がけるつもりではありますけど)一応、こういうときの対応の方法を載せておきましょうね。とりあえず、バーフォンの違うBASICで打ってしまったらASCIIセーブしてください。

SAVE「ファイル名」,A
それからリセットして本来使うはずだったBASICを立ち上げます。そして、再びロードすればOKです。

おな、そうで。ショートプロで質問の多かったものに5月号のDNAデモがあるんですが、これはコンパイル時のスイッチを小文字にしてしまった人が多かったみたいです。コンパイルできなかった人はそこを注意してもらう一度やってみてください。リストにバグはありません。

うーむ、なにやら「あの筋?」質問箱になってしまった。

☆(で)のばーいハNZ(その3)はものすごーくうれしい。

(アンケートハガキ、白井達広さん)
ありがとー。私も本当にうれしいです。あのハNZって結構大変なんですよ。なにしろOh!Xには珍しく毎月ちょっとずつプログラムを載せていて形式なんて始める前の下準備がめんどくさいわ。文字が小さいから1ページでショートプロ3ページ分の文章を書かなくちゃいけないわけ、いや、よかったよかった。延長戦もよろしくね(あと、リクエストもね)。

うーん、アンケートハガキっていいなあ。と思ってるこんなハガキもあったりします。

☆(で)のばーいハNZのコーナーを3ページくらいに増やしてほしい。

(アンケートハガキ、箕浦健一郎さん)
——さんべんしてよ(でもうれしい!)。ま、なにはともあれ、これからもよろしくお願いします。

恵まれている(で)に愛の手を!

さて、さてさて。結構のんびりやっていたはずのこのコーナーもいよいよ今月の敵と敵のタマの動き、そして来月の当たり判定を残すのみになっちゃったんです。ということで本来なら来月で「それではみなさん、さよーならー」となるはずだったのですが、皆様のハガキのおかげで再来月からはぱーていハンスは第2部に突入することとなりました。はい、拍手拍手! でも、連載が延長になるのはうれしんだけど、まさかこうなるとは予想すらしてなかったんで、はっきりいってまだなにをどうするのか全然準備してなかったりするのですよねー。困ったなー、急になんか作れっていわれてもなに作っていいのかわからんよー。てなわけでこんなものを作ってほしいとかこうしてみてもどうかとか、こういうところがわからなかったとかいうハガキを大募集しちゃいます。ネタのない(で)にあいの手をー! あーこりゃこりゃ(そのあいの手じゃなーい)。

敵襲だーっ! ゲームの個性だーっ!

さて。というわけで、敵の出現、敵の移動、敵のタマ撃ち、敵のタマの移動です。

シューティングってゲームセンターにもいろいろなのがありますが、基本的には自機を動かしてタマを撃つという意味でそんなに変わらないですね。シューティングゲームの個性って敵の出現、動きのパターンや背景なんかがかなりの部分を占めていると思うんです(例外も多々ありますが)。だからシューティングゲームを作るとき、背景をカラフルにしたり、デカキャラを作ったり(X68000だったら簡単でしょ)、敵キャラの動きをなめらかにしたり、あと、敵のタマが多くなりすぎてバランスが悪くなってしまわないようにとかの努力をすれば市販ゲームぐらいにできなくはないと思うんですよ。特に、X68000みたいにスプライトやBGがあったりなどなどという機能が揃ってるマシンだとアフターバーナーみたいに特別なプログラムテクニックが必要なものでない限り、アマチュアで作ったゲームと売ってるゲームの差は、極端にいうとどれだけデータを作れるか(どれだけの人々がどれだけ時間をかけたか)、どれだけ妥協しないで作ったかの差ではないかと思います。ゲーム作りの極意は根性(もちろん創意工夫も)なんです。決してテクニックだけではありません。

皆さんにはそのようにがんばっていただきたいなー、ということで今回私は手を抜かせていただきます(い、いままで並べたゴタクはいったいなんだったんだ……)。

で、敵の動きなんですがとりあえずこんなのを考えてみました。

「敵がすーっと下りてくる」

「ばっ、とタマをまき散らす」

「敵はすーっと逃げていく」

なんかとんでもなくいやな性格してる敵キャラですけどねー。んで、敵をとりあえず動かす

てみたいんですけど、その前にちょっと思い出さなきゃいけないことがある。そうそう、先月いったあれなんです。自機も敵も同時に動かさなくちゃいけないので、かわりばんこで動かすように組んでやらなくちゃいけないんですよ。先月、自機とタマを交互に動かすために自機のメインルーチンの中に、

```
firemove()
```

って1行入れて自分のタマを動かすルーチンと呼び出してましたよね。それと同じように、

```
enemy_move()
```

って1行入れて敵を動かすルーチンと呼び出してやるんです。ちなみに敵のタマを動かしてやるルーチンが、

```
bomb_move()
```

なんですけど、敵のタマももちろん同時に動くわけですよ。だからbomb_move()もそこに入れていい……んですが、なぜかbomb_moveはenemy_moveが呼び出しています。別にこれは意味はないんです、っていうか実はなんでこうしたのかよく憶えてないんです(こらこら)。たぶん敵が動くルーチンと敵のタマを動かすルーチンだけほかのルーチンと別の日に作ったので思わずそうしてしまったんじゃないかな。別に次々とルーチンがルーチンを呼んでもかまわないっちゃかまわないんですが、やっぱりリストが読みにくいですからみなさんはちゃんとどっからかに統一しましょうね。

それはそうと敵が出てきて引き返す(折り返す)っていうほうがわかりやすいかな? ってえことは、まず、敵がどこで折り返すか決めておいて、それから敵をつつーと下ろしていった、折り返し位置にきたら帰っていくようにすればいいわけですね。さて、ここで問題です。ここではいくつ変数を作ればいいでしょう。

自分のX座標、およびY座標

折り返し点のY座標

自分が上がっているか下がっているかのフ

ラグ

うん、4つもあればよさそうですね。自分が上がっているか下がっているかのフラグはたとえば、

上がっているとき=-1

下がっているとき=+1

としてやれば敵を動かすときに(たとえば敵のY座標がenemy_y、フラグがenemy_sgnという名前だとしたら)、

```
enemy_y = enemy_y + enemy_sgn
```

としてやればできそうですね。

それじゃ、1つひとつルーチンを作っていきますか。まずは敵の出現。

```
enemy_appear()
```

とりあえず、

「タマは出さないか」

「自分のX座標と引き返し座標を決める」

「上がり下がりフラグを+1にする」

このくらいかな。で、これを敵のY座標が0の(つまり敵が現れない)とき、このルーチンと呼んでやればいいわけね。んで、

```
enemy_move()
```

出てきた敵をこのルーチンで動かす。これは敵を順番に1ステップ動かすわけですね。んで折り返し点にきたら上がり下がりフラグを1にしてタマを出させます。

```
bomb_move()
```

タマが出ていたらタマを1ステップ進める。で、「タマをばらまく」ことにしたわけですが、とりあえずタマは3つ出して左下、下、右下に進めます。

あー疲れた。とりあえずこんなもんかなー。

さて、来月は当たりチェックやおしまいね。んー、でも当たりチェックだけで1ページもたすの苦しそうだなー(たぶん1/4ページくらいで終わっちゃうと思うんだよね)。ま、いいか。明日は明日の風が吹くと。来月またこのOh! Xで。ガガガガ(と穴を掘って去る)。

```
330 enemy_move()
330 /* 敵の動き*/
340 func enemy_move()
350 for i=0 to 2
360 if enemy_y(i)>enemy_b(i) then enemy_sgn(i)=-1:enemy_fire()
370 if enemy_x(i)>0 then enemy_y(i)=enemy_y(i)+enemy_sgn(i)*8
380 j=0:for a=0 to 2:j=j+bomb_y(i,a):next
390 if j=0 and enemy_y(i)=0 then enemy_appear()
400 sp_set(38+i,enemy_x(i),enemy_y(i),{(enemy_sgn(i)-1)*2}*#H8000+#H123)
410 next
420 bomb_move()
430 endfunc
440 func enemy_appear()
450 enemy_b(i)=0:while (enemy_b(i)=0 or enemy_x(i)<16 or enemy_x(i)>193)
460 enemy_x(i)=rand()and#HF0:enemy_y(i)=8:enemy_sgn(i)=1
470 enemy_b(i)=rand() and #HF0
480 endwhile
490 endfunc
500 func enemy_fire() /*敵もタマを撃つ*/
510 for j=0 to 2
520 bomb_x(i,j)=enemy_x(i):bomb_y(i,j)=enemy_y(i)
530 next
540 endfunc
550 func bomb_move() /*タマの動き*/
560 for i=0 to 2
570 for j=0 to 2
580 if bomb_y(i,j)>0 then bm_sub()
590 next
600 next
610 endfunc
620 func bm_sub()
630 bomb_y(i,j)=bomb_y(i,j)+8
640 if j=0 then bomb_x(i,j)=bomb_x(i,j)-8:if bomb_x(i,j)<0 or bomb_y(i,j)>256
then bomb_y(i,j)=0
650 if j=1 and bomb_y(i,j)>256 then bomb_y(i,j)=0
660 if j=2 then bomb_x(i,j)=bomb_x(i,j)+8:if bomb_x(i,j)>192 or bomb_y(i,j)>256
then bomb_y(i,j)=0
670 sp_set(42+i*3+j,bomb_x(i,j),bomb_y(i,j),#H122)
680 endfunc
```


X68000用 OMENS OF LOVE

X1/turbo用 ENDLESS RAIN

X68000用MUSICDRVサンプル曲 ©NAMCO

ダートフォックスより **Running up!**

Kodama Kazuhiko
小玉 和博

Fushiki Yoshihiro
伏喜 義宏

Nishikawa Zenji
西川 善司

サンプリングは使用していません

X68000用に「OMENS OF LOVE」をお届けしましょう。この曲はフュージョンと呼ばれるジャンルの曲で、T-SQUAREが演奏しています。T-SQUAREは5人のグループで、カシオペアと並んで日本が世界に誇れるフュージョンバンドです。F1グランプリの曲、「TRUTH」などでおなじみですよ。

曲はインストなので、比較的FM音源だけのコンピュータでも作りやすい構成とは思いますが、テクニック命といっても過言ではないフュージョンを完全に再現するのは、かなり厳しいのではないのでしょうか。特にFM音源とは相性が最悪ともいえるようなギターが前面に出ている曲は至難の技だと思えます。

さて、作品のデキはといいますと、とっても気持ちいい曲になっています(?)。FM音源のみでAD PCMを使っている普通のOPMDRV.Xのみで演奏できますが、サンプリングドラムに頼らなくても立派に演奏できるというお手本のような仕上がりです。もともとOPMAではボスコニアのサンプリングデータを使用していま

したので、あの元気なドラム達そのまま使われていました。そうすると静かな曲や、落ち着いた曲などではどうしてもドラムだけが浮いてしまっていたのです。そういった意味でも、この曲ではOPMのみで演奏したほうがきれいになるのです。試しにサンプリング対応にしてみたところ、やはりドラムが浮いていました。納得できない人は自分で試してみてください。

そういえば、フェードアウトもOPMだけならきれいにキマるといってもいいかもしれません。

XシリーズのX

X1用にはXの「ENDLESS RAIN」をお届けしましょう。Xはライブハウスからの叩き上げバンドです。自主制作していたアルバムが2枚あって、それがバカ売れしたためレコード会社の目にとまり、プロデビューに至ったという経歴を持っています。かなり正統派のバンドといえるでしょう。残念ながら矢板にあるSHARPさんのお抱えバンドではありません、悪しからず。

さて、作品についてですが、なかなか面白い構成をしているのではないのでしょうか。ヴォーカルをギターの音でやってしまったわりには、かなりまとまりがよいといえます。前述のとおり、ギターの音は結構ムズいのです。その分を考えるとよくできているといえます。

惜むべきこととして、曲調を考えるとちょっとドラムの音が大きいのでは? と思えます。特にPSGのハイハットが怒鳴っています。確かに、原曲でははっきりと聞こえてはくるのですが……。ハイハットはノリを出すのにも使われますが、曲を引き

締めるのにも使われます。おそらく原曲の使い方は後者でしょう。引き締めるためのハイハットが、全体的に繊細な音で構成しているのを壊してしまうのはちょっともったいないですね。



締めるのにも使われます。おそらく原曲の使い方は後者でしょう。引き締めるためのハイハットが、全体的に繊細な音で構成しているのを壊してしまうのはちょっともったいないですね。

やはりFM音源と比べてPSGの音質が落ちてしまうのは仕方ないことですので、PSGの使い方はしっかりと考えてみましょう。ソフトウェアエンベロープを掛けてコーラスラインとか、ハイハットならポリュームを小さめにするとか、S.E.を作ってみるなどが挙げられます。ミキサーをつないでいる人は、PSGの音量をFM音源の7割程度にしてみてください。あとは、好みに合わせてドラム系の音を心持ち下げて聴いてみてください。(S.K.)

「MUSICDRV」用サンプル曲

「MUSICDRV」用のサンプル曲のプログラムとして(注意:「OPMD」では演奏できません)。ナムコのトップビュータイプのアフロードカーレースゲーム「ダートフォックス」のメインテーマ「Running up!」をお届けします。この曲は、カシオペアの「Looking up」のパロディともいえる曲で(名前まで似ていたりする)、フュージョン



T-SQUARE

風のアレンジとなっていますからそっちの筋の方にもおすすめ。なんといってもウリはチョッパーの効いたベースと耳に突き刺すような高音のシンセソロ、右左にパンするパッキングです。作曲はもちろん(?)「メタルホーク」のめがてん細江氏です。

演奏方法

対応楽器はM1/R/Tシリーズ(以下M1)専用です。M1とMT-32の両方をお持ちの方はそのシステムに対応します(MT-32のみでは演奏できません)。また、FM音源も使用しているため、ミキサーなどでミキシングしてお楽しみください。まず、演奏させる前にM1側の設定をしてやります。

0) 「MUSICDRV」を、

A>MUSICDRV #180

のように組み込んでください。

1) 曲中で使用されている音色のベンドレンジを変更してください(2, 4, 46, 48, 51, 71, 72, 75, 92:後述の「MIDI基本テク特集」で説明してあります)。

2) M1リズムキット(音色番号09, 49)をEDIT PROG, F4-3「VDA1 KBD TRK」のパラメータを図Aのように設定します(リズムの音色をほかの音色よりも音量をやや大きくするため)。

3) GLOBALモードにしてDRUM KIT3の「TOM2」をすべて「TOM1」に、「OPEN

HH」と「CLOSED HH」のPANを(9:1)に変更してください(図B)。

4) 次にシーケンサモードにして、F1-4「MIDI CH」で各トラックのMIDIチャンネルを1, 11, 12, 13, 14, 15, 16, 10のように設定してください(図C)。

5) F4-1「TRACK PARAMETER」で各トラックのパンポットを(5:5), (5:5), (9:1), (1:9), (5:5), (8:2), (5:5), (5:5)のように設定してください(図D:プロテクトオフにしてから設定すること)。

6) MT-32もお持ちの方は、M1のMIDI THRU 端子から MT-32の MIDI IN へ MIDIケーブルを接続し、MT-32の電源を入れてください。M1のみをお持ちの方は特に接続する必要はありません(当たり前だ)。

7) メインプログラムを入力、または入力されたものをロード、RUNしてください。
注意:MUSICDRVは7月号のデバッグ(バグを取ること)を行ったものを使用してください。さもないと、FM音源の音色が正常に鳴りません。

テクニックの解説

特に変わったことはしていませんが、ダンパーとピッチベンドを多用しています。ピッチベンドのMMLデータはbnd()という関数で作っています。たとえば、
bnd("c",12,8192,8875)

図A リズムキットの設定 1

PROG I49 VDA1 KBD TRK Center Key
C-1 A+00 EGtime=0 AT:0 DT:0 ST:0 RT:0

図B リズムキットの設定 2

DRUM KIT3 Closed HH1
#05 11 F#1 +009 L-57 D+00 9 1

図C MIDIチャンネルの設定

SONG1 MIDI CH
1G 11 12 13 14 15 16 10

図D パンポットの設定

SONG1 TRACK PARAMETRE
Tr1 I01 V99 T+00 D+00 5 5 Prot: OFF

は、12個分の精度でピッチ8192(C)からピッチ8875(C+)まで滑らかに変化させるMMLを生成します。値、用語の意味については後述の「MIDI基本テク特集」を参照してください。

ところで、M1はコントロールチェンジにパンがありません。そのため基本的にはリアルタイムに音をパンすることが不可能です。しかし、各トラックにあらかじめ適当なパンを設定しておき、MIDIチャンネル切り替えコマンド「@n」で演奏チャンネルを切り替えることにより、パンをリアルタイムに切り替えているようなニュアンスを出すことができます。初心者の方のM1ユーザーは参考にしてください。(善)

Oh!X通巻100号記念 MIDI基本テク特集

私が、サングラスをかけるとほとんどチンピラの西川善司です。6月号の創刊8周年記念のディスクに付いてきた「OPMD.X」と「MUSICDRV.X(サン・ミュージカル・サービス)」ともに好評だったようです。両ツールともに、MIDI楽器をFM音源感覚のMMLで演奏可能なため、MML派の人間にとってまさにたなからボタもちでしょう。

しかし、MIDI楽器はMIDI楽器。細かな表現をするのに大変重宝していた「Yコマンド」が使えないのははじめとして、MIDI楽器はFM音源やPSGとは違った箇所が多くあります。そこで、この場を借りてMIDI楽器を使うにあたっての基本テクを、音楽特集でもないのにババーンと公開してしましましょう。

MIDI楽器でディチューンをやる

FM音源(OPM)では、Y48+チャンネル番号(0~7)でピッチ(音程)を微妙にずらした音を重ねてやることによって、コーラス効果を実現できました。MIDI楽器には、こういったピッチをずらすコマンドはないのでしょうか。「ピッチ」という言

葉でピンときた読者もおられるでしょう。そうです、「ピッチベンド」のコマンドを用いるのです。

「MUSICDRV」では@B8192がピッチの基準値です。1チャンネルはこの基準値で鳴らしてやり、もう1チャンネルは基準値@B8192±50~100程度で鳴らしてやります(FM音源部もこの方法でコーラス効果を実現してやることができます)。

「OPMD」では基準値は128ですので1チャンネルは「Y9, 128」で鳴らし、もう1チャンネルは「Y9, 128±1~3」程度で鳴らしてやりましょう。

ここで注意がひとつ。楽器によってベンドの範囲が違うという点です。ROLAND MT-32は初期状態で1オクターブ範囲のピッチベンドが可能です。KORG M1/R/Tシリーズ(以下M1)では初期状態では半音範囲です。でもご安心を。たいいていの機種は、このピッチベンドの範囲についてはコンフィギュレーションが可能です。M1の場合は音色単位でこの設定が可能です。音色を呼び出したあと、「EDIT PROG」モードにし、F7-2「JOYSTICK」(図1)の「P+02」を「P+12」にすることにより、MT-32のような1オクターブ範囲のピッチベンドが可能となります。そうそう、パラメータを

書き換えたあとはF9-1「WRITE/RENAME」(図2)で音色を再登録しなければいけません。まあ、ピッチベンドは1オクターブ範囲にしておいたほうが音色の応用範囲が広がるので、M1ユーザーはすべての音色をいまいった方法で変更しておきましょう。

ちなみに、ベンドの範囲を1オクターブにしたとき、「MUSICDRV」では半音が±683、「OPMD」では±11となります。つまり、「MUSICDRV」で、
@B8192C@B8875C (8192+683=8875)

図1 JOY STICK

PROG I00 JOY STICK
P+00 F+00 PM00 MF0 FM00 MF0

図2 音色登録

PROG I00 A.PIANO Write/Rename
[<] [>] [WRITE] → I00

とすると、最初の「C」はもちろん「C」ですが、2回目の「C」はC+で発音されます。「OPMD」で同じことをするには以下ようになります。

Y9,I28CY9,I39C (I28+I1=I39)

上の値をもっと細かいステップで与えて、各音を「&」でつないでやることにより「ポルタメント」を表現できます。

＆のお話

読者のハガキのなかにこんなのがありました。「OPMDでC&C+とする、とCの発音後C+の音が同時に鳴ってしまいます」。

FM音源では上のようにすると、Cの発音後、C+のアタック音なしに音程がC+へと変化します。「OPMD」では「&」はキーオフしないという目印に過ぎません。ですから、次にきたC+はCをキーオフせずに鳴ってしまいます。FM音源では1チャンネル1声という大原則があるので問題は無いのですが、MIDI音源は1チャンネルで和音も発音可能なので、ハガキにあるような現象が起こるので

す。手抜きというより「OPMD」の性質上仕方ない現象なのです。しかし、後述の「ダンパー」効果で、この現象を逆手に取って実現できます。

ところで「MUSICDRV」では、ハガキにあるような例を演奏させると、

C+C+

のように「&」が削除されたようなMMLが演奏されます。つまり以前に鳴ったキーコードとは違う音が新たに発音される場合、以前に鳴っていた音は強制的にキーオフされるわけです。また、FM音源部においても同様の処理が行われるので注意が必要です。

では、「MUSICDRV」や「OPMD」で「タイ」や「スラー」を実現するにはどうしたらよいのでしょうか。答えは簡単。先ほど、説明した「ピッチベンド」のコマンドを使ってやればよいのです。

@B8I92C&@B8875C MUSICDRV

Y9,I28C&Y9,I39C OPMD

おわかりいただけたでしょうか？

ダンパーってなんだー

「MUSICDRV」では「@d」というコマンドがあります。これは、「ダンパー」という機能を「オン/オフ」するもののなのですが、いままでFM音源のMMLのみを使っていた人にとっては耳新しい言葉です。言葉で説明するより例を用いて説明したほうがわかりやすいので、実際に「ダンパーコマンド」を用いて楽器を演奏させてみることにしましょう。

L16R@d127CEGR2.@d0

をいま演奏させたとしましょう。「L16」はただのデフォルト音長設定、続く「R」は16分休符となります。次の「@d127」はダンパーオンのコマンドで、以後発音される音はダンパーオンの効果がかかります。最初のCが発音され続いてEが発音されますが、このときCの音はキーオフされません。同様に最後のGもCとEが鳴った状態で発音されます(つまり、この時点ではCEGの和音が鳴っている)。さて、次に付点2分休符である「R2」がきています。通常だと無音状態となるのですが、ダンパーオンの影響で「R2」の時間、「CEG」の和音が鳴り続けます。そして、やっと最後の「@d0」

でダンパーオフとなり、発音されていた音はすべてキーオフされます。この例は譜面にするとちょうど図3のようになります。

「OPMD」でこれをするには「&」を用いてやります。説明は「&」のところでしたので省きますが、上の例は「OPMD」では、

L16RC&E&G&G2.

となります。ただ「G2」のあとにオールノートオフのメッセージを送らないと、CとEの音が鳴りっぱなしとなるので注意。

ベロシティのお話

これまた、FM音源から入ってきた人には耳新しい言葉です。MIDIの専門書などには「音の立ち上がり方の速さ」などと書いてありますが、「ポリウム」のことだと思ってくださって結構です。いい方を変えれば「どのくらいの強さで鍵盤を叩いたか」ということです。ですから、実際の音量はVコマンドの値×このベロシティの値で決定されます。「MUSICDRV」で最大の音量で演奏するには、

@v127 @ul127

を最初に送ってやります。

また、リズムマシンやポータブルキーボードのなかにVコマンドを認識しない機種があります(YAMAHA RX-8など)。そういった機種に対してはこのベロシティのみが音量を決定します。

@Lのお話

「MIDIドライバで@Lなどの微小音長を多用すると遅くなります」といった内容のハガキが届きました。うーん、FM音源を酷使する人は@LでガリガリとMMLを書く人が多いようですね。そういえば、常連の立川正之君などは8分音符以上の音長は減多に書きません、なんて言ってました……。MIDIは31250bpsという、速いようで実はそんなに速くないボーレートで通信をしています。私の貧弱なMIDIシステムでも発音遅れはよくあることです。@Lの多用でテンポが遅れるなんてことは当然といえます。こういった問題の解決策としては、

- 1) @Lの使用を少し控える。
- 2) 内蔵FM音源に対してのみ@Lを使用する(内蔵FM音源は、MIDIで通信をしているのでなくI/OでMPUと直結しているため、かなり高速な応答が可能です)。
- 3) 適当なトラックのMMLの最初に「@LIR」を挿入し、割り込み周期のずれを作ってやる。

が挙げられます。

「MUSICDRV」のバグ

「MUSICDRV」にはバグがいくつか発見されています。「MVSET」コマンドで設定した音色番号とMMLの「@」コマンドの番号と対応しない、というバグは7月号で訂正されていますが、ほかのバグは取る手立てがないため(制作はサン・ミュージカル・サービスで、ソースリストはOh!X編集部にはありません)、これから話す解決方法で対処してください。

和音のコマンドは、

'CEG'

のように「」内の音を同時に鳴らすもので、最初の音に音長を書けばその長さで和音が鳴り続ける

図3 ダンパーの譜面



という、いままでのMMLの常識を破った大変便利なコマンドです。しかし、

CEG

のように付きを含む音長を記述すると、暴走してしまいます。これは、デフォルト音長設定コマンド「@L」を使ってやれば簡単に同様のことができます。つまり、付点3分音符なら、

@L36 CEG

です。では、全音符を超えた音長で和音を鳴らし続けるにはどうしたらよいのでしょうか。「@L」では全音符である132以上は記述できません。

答えは「ダンパー」を用いて以下のようにしてやります。

@v127 @EGRIRI@d0

この例だと、全音符3個分の長さで和音CEGが鳴り続けます(原理はすでにダンパーのところで説明したのでここでは省略します)。

「@v」はMMLトラックをMIDIチャンネルいくつに割り当てて演奏の途中で切り替える大変便利なコマンドです。「OPMD」では「Y4,?」、「Y5,?」にあたります。「MUSICDRV」で以下のようなMMLを書いた場合、正しく動作しないので注意が必要です。

@v1 @v2 @v3 @v4 @v5 @v6 @v7 @v8 @v9 @v10 @v11 @v12 @v13 @v14 @v15 @v16 @v17 @v18 @v19 @v20 @v21 @v22 @v23 @v24 @v25 @v26 @v27 @v28 @v29 @v30 @v31 @v32 @v33 @v34 @v35 @v36 @v37 @v38 @v39 @v40 @v41 @v42 @v43 @v44 @v45 @v46 @v47 @v48 @v49 @v50 @v51 @v52 @v53 @v54 @v55 @v56 @v57 @v58 @v59 @v60 @v61 @v62 @v63 @v64 @v65 @v66 @v67 @v68 @v69 @v70 @v71 @v72 @v73 @v74 @v75 @v76 @v77 @v78 @v79 @v80 @v81 @v82 @v83 @v84 @v85 @v86 @v87 @v88 @v89 @v90 @v91 @v92 @v93 @v94 @v95 @v96 @v97 @v98 @v99 @v100

順番を追って説明すると、まず、@v1で現在演奏中のトラックをMIDIチャンネル1に変更し、次に音色切り替え「@v3」で音色が切り替わります。「@v3」は、まあ、MML演奏データがずらーっと並んでいるとして、これらはすべてMIDIチャンネル1で音色番号3で演奏されます。

さて、次に「@v2」がきているのでトラックをMIDIチャンネル2に変更します。問題は次の「@v3」で、なんとMIDIチャンネル2へ音色切り替えのメッセージが送られないのです。どうも同じ音色は切替えないというアルゴリズムのもとで動作しているらしく、しかもそれをMIDIチャンネル単位でなくトラック単位で行っているため、このような現象が起こるのでしょう。この対処方法としては、音色切り替え専用のトラックを設けたり、別のトラックに音色切り替えのコマンドを挿入する、などが考えられます(ちょっと空しいね)。

いずれにせよ、サン・ミュージカル・サービスさんの迅速な対応が望まれますね。

MUSICDRVに望むこと

「MUSICDRV」はとてもよくできています。欲をいうと、ピッチベンドはオートベンドにしてほしい。また、ベロシティやダンパー、モジュレーションなどの頻繁に使うコマンドは、できたら「@」という文字の必要のない1文字コマンドにしてほしいかったです。あと和音のコマンドはFM音源にもほしいなあ。サン・ミュージカル・サービスさん、Ver.2に期待してます。ゴロニャー。また、何かMUSICDRV楽器について質問があればどうぞ、Oh!X編集部・西川善司まで。


```
960 a="07005184.>b4a4gsg2refg4.f.f2e4.d4e4c2b<cde4.ba4ed4  
ldrlc1d4.>b4a4gsg2refg4.f.f2e4.d4e4c4":m_trk(1,<@127y50,ddd")  
.m_trk(5,"v12r15y52,,28+a)  
970 a="ccccccc>eeeeeeeffffffffgggggg+g+g+aaaaaaa<ddddddd>  
gggggggggv13f4ecrd4.vl0cccccc>eeeeeeeffffffffgggggg+g+g+":m_  
trk(4,a)  
980 a="l1@74o4vl4edcd2d2edc@70v13cf4e8c8r8d4.>@74vl4edcd2d2":m_  
trk(6,a)  
990 a="l1@74o4vl4c bab2b2(c)>aa@70v13<<c4c8>g8r8(c4.)>@74v14c>ba  
b2b2":m_trk(7,a)  
1000 a="l1@74o3v14ggfg2g+2af+f@70v13<g4g8g8r8g4.>@74v14ggfg2g+2  
":m_trk(8,a)  
1010 /z  
1020 a="aaaaaaaa<ddddddd>ggggggggggggggff4fffffffffffeeee  
eeaaaaaaa<dddddddggggggggga&a2aeaeab>aaa(e)a":m_trk(1,a)  
1030 y="o2@76a<@75e>@76a<@75e>":a=y+y+y+y+y+i+m_trk(2,z+z+  
"+o2@76a<@75e>@76a<@75e8>@76a8&"):m_trk(2,a)  
1040 a="<?>b<cde4.d4c4ag&l&g2redc&c2c4ed&d2d4g&glg2r4rggf4er  
fre4.d4c4de&l&e2r4re"  
1050 m_trk(3,a):m_trk(5,a)  
1060 a="aaaaaaaa<dddddddgggggggggggggg4fffffffffffeeee  
ee>aaaaaaaa<ddddddd>ggggggggg<ekel&l":m_trk(4,a)  
1070 a="edc2&c8c4.>b2d4.&@73v13c8&cddc+cc2">a-4.<c+8&c+c+>":m_tr  
k(6,a)  
1080 a="<c>a2&a8r4.g2b1.@73v13a8&abbaa2f4.a8&a&a":m_trk(7,a)  
1090 a="af+f2&f8r4.r2r4.&v127@a8a2a2d2dg4.>b8r8d4.c+c+8c4.c  
lcc4.eb4a4r4>b4c4>a4.<  
1100 m_trk(8,a)  
1110 /z  
1120 <">dddddddeeeeeee>eeeeeeeea4.ag4.gfffffffeeeeeee<ddd  
dd>g4ggggggg4gggr4rc&  
1130 m_trk(1,d)  
1140 ezy+y+y+y+y:m_trk(2,e)  
1150 f="o2@76a<@75e>@76a<@75e8>@76a<@75e8e8e>@76.a8&":m_trk(2,  
f)  
1160 g="f4efrag&g2.rgg+3abrer<c&c2rccoc4>barbr<c2>g4gcef4edrc  
rd&ld&2refg&  
1170 m_trk(3,g):m_trk(5,g)  
1180 h="dddddddeeeeeee>eeeeeeeeaaggggggfffffffeeeeeee<ddd  
dd>g4ggggggg4gggr4.<c&":m_trk(4,h)  
1190 j="c">ggg+naga2..&g&g2..&g&g2r4.v14l8&":m_trk(6,j)  
1200 k="aeefeef2..e&e2..e&e2&r4.<v14l8c&":m_trk(7,k)  
1210 l="l1f4.frf4.e4.drc4.e2e2a2a2a4.ara4.g4.grg4.a4.a2gd4c8g4  
&g4.<@73v14r4.g&":m_trk(8,l)  
1220 /z  
1230 a="cccccc>e4eeeeeeegggggggggggggggaaaaaa<d4ddddd4c+c+c+c+c  
+c+cccccc":m_trk(1,a)  
1240 m_trk(1,i)>bbbbbegg<c&":m_trk(1,a)>i>bbbbbeea&a2..g+g&+2..&g  
g2..f+f&+2..<f&")  
1250 for i=1 to 2  
1260 a="o2@76a8a8<@75e>@76a<@75e8>@76a4 a8<@75e>@76a<@75e":m_tr  
k(2,a)  
1270 a=y+"o2@76a<@75e>@76a<@75e8>@76a a8<@75e>@76a<@75e8>@76a  
a8<@75e>@76a<@75e>"+y:m_trk(2,a)  
1280 a="o2@76a<@75e>@76a<@75e8>@76a8&"  
1290 b="@76a<@75e>@76a<@75e8>@76a8&|4ar4<@75d>r8@76a8&:  
1300 if i=1 then m_trk(2,a) else m_trk(2,b)  
1310 next  
1320 a="g2refg&g2.r4r>b-<dargrf4fer4def&f2rdef&f2.r4r>a<cgrfr&  
"  
1330 b="eedr4efg&":c="eedr4edc&c2r>b<cd2.ede2.d+ea4.e4dcrf&  
1340 br<(dr,c)+a+b+a+c":m_trk(5,a+b+a+c)  
1350 a="cccccc>e4eeeeeeegggggggggggggggaaaaaa<d4ddddd4c+c+c+c+c  
+c+cccccc":m_trk(4,a)>+>bbbbbegg<c&"+a">+>bbbbbeea&a2..g+g&+2..&g  
g2..f+f&+2..<f&")  
1370 a="eedrrr4dd4gdr4.@74@v127dlc+2@73v14r4.f4ferf4f4farf4.  
@74@v127f1  
1380 m_trk(6,a)+"2g4.@73v14e&"+a+"d2b4.@73v14r4>a&ea&2rcf<c&c2>  
reg<c&c2>rea(e2)v14c&"  
1390 a="<c>br<cr4>b4b<dr>br4.@74@v127b-1a2a4.<v14@73d4dc+rdrfcl  
ofcr4.@74@v127c1  
1400 m_trk(7,f)+"2g4.@73v14c&"+a">b2g+4.v14a&a2..g+g&+2..&g&g2..  
n&a&2..@73v13a&")  
1410 a="gggrgrg4gbgrg4.@74@v127gle2e4.@73v14a4aararra4a<c>rar4  
.@74@v127al  
1420 m_trk(8,a)+"b2<d4.>@73v14g&"+a">g+2@c4.v14&e2..f+f&+2..&e  
2..f&f2..n&")  
1430 /z  
1440 a="fferderf4ferd>gr<c4ccc1:12ccc:lccc>g&gl<cccccc>eeee  
ee":m_trk(1,a)  
1450 a="14a8a8<@75e8>@76a8r8a8&a<@75e8>@76a8r8a8&"+z+z+?"@76  
a<@75e116efdcl4>@76a":m_trk(2,a)  
1460 a=z+z+?"@76a<@75e>@76a8a4.r2<@75e>":x="o2@76a<@75e>+r8@76a8  
a<@75e>":m_trk(2,a,x+x)  
1470 a="l1fferdcfr4ferderc&l3c2.c8r8 @74v13o4rfederf4fef4cdrd  
4.drgrf&f2.ef4federc&a4 v15@72y50,8rgbg<dd+32&e8..&d+32&d16..&  
2&ge&ab<dk&g":m_trk(3,a)  
1480 a="fferderf4ferd>gr<c4ccc1:5cccc:jcv13<rcce>v10cc1:5cccc;  
jcccd&dlcccccc>eeeeeee  
1490 m_trk(4,a)  
1500 a="l18fferdcfr4ferderc&l3c2.&c8. v13@74o4rcc4>b<rc4ccc4>b<rc4cc4>  
br<c4>.>br<droc&2.cc4cc4>b<rc4a4 v15@72y52,32rgbg<dd+32&e8..&d  
+32&d16..&g2&ge&ab<dk&g":m_trk(5,a)  
1510 a="<c>gragr<c4>graaro4@70v13c2drgrf&f2.ef4ef4derf4fec4cdr  
e4.drgrf&f2.ef4federc&l4v14o4@71lled  
1520 m_trk(6,a)  
1530 a="aaferf4aerffro4@70v12c4a4.>br<droc&c4.c4cc4a4>b<rc4cc4>  
a>br<c4>.>br<droc&2.cc4cc4>b<rc4a4 v15@72y52,32rgbg<dd+32&e8..&d  
+32&d16..&g2&ge&ab<dk&g":m_trk(7,a)  
1540 a="aagragra4agraa ro3@70v12g&g4.grbra&a2.ga4ag4ggra4ag4grg4.  
grbra&a2.ga4ag4ggrf4llio3@74v14l1gg":m_trk(8,a)  
1550 /z  
1560 a="fffffffgggggg+g+g+g+aaaaaaa<ddddddd>ggggggggg4rg4.<  
cccccc>eeeeeeeffffffffggggggg+g+g+":m_trk(1,a);m_trk(2,x+x+x+x  
+x+x+x+x+x+x)
```


リスト5 Running up I

```

10 /* **** */
20 /*
30 /* Running up */
40 /*
50 /* (C) NAMCO / MEGATON
60 /*
70 /* Arranged by Z.NISHIKAWA
80 /*
90 /* **** */
100 m_init()
110 dim char v(4,10)
120 /* AF OM WF SY SP PMD AMD PMS AMS PAN S.E.1
130 v=[59, 15, 2, 1,200,127, 0, 0, 0, 3, 0,
140 /* AR DR SR RR SL OL KS ML DTI DT2 AME
150 31, 8, 1, 8, 7, 20, 2, 1, 5, 3, 0,
160 31, 8, 8, 7, 5, 24, 1, 2, 1, 1, 0,
170 31, 3, 7, 8, 1, 21, 1, 1, 3, 0, 0,
180 31, 0, 0, 9, 0, 0, 2, 8, 5, 2, 0]
190 m_vset(1,v)
200 /* AF OM WF SY SP PMD AMD PMS AMS PAN S.E.2
210 v=[56, 15, 2, 1,100,127, 0, 4, 0, 3, 0,
220 /* AR DR SR RR SL OL KS ML DTI DT2 AME
230 20, 10, 0, 5, 0, 44, 8, 0, 0, 0, 0,
240 20, 16, 0, 10, 15, 27, 0, 0, 0, 0, 0,
250 20, 10, 0, 6, 15, 9, 0, 1, 0, 0, 0,
260 31, 18, 0, 10, 15, 0, 1, 13, 0, 0, 0]
270 m_vset(2,v)
280 /* AF OM WF SY SP PMD AMD PMS AMS PAN S.E.3
290 v=[59, 15, 2, 1,200,127, 0, 7, 0, 3, 0,
300 /* AR DR SR RR SL OL KS ML DTI DT2 AME
310 31, 8, 1, 8, 7, 40, 2, 2, 5, 3, 0,
320 31, 8, 8, 7, 5, 36, 1, 2, 1, 1, 0,
330 31, 3, 7, 8, 1, 29, 1, 14, 3, 0, 0,
340 31, 0, 0, 9, 0, 2, 2, 8, 5, 2, 0]
350 m_vset(3,v)
360 /* AF OM WF SY SP PMD AMD PMS AMS PAN GLOCKEN
370 v=[52, 15, 2, 1,200,127, 0, 0, 0, 3, 0,
380 /* AR DR SR RR SL OL KS ML DTI DT2 AME
390 26, 3, 0, 2, 15, 35, 1, 6, 3, 0, 0,
400 31, 6, 0, 6, 15, 0, 1, 2, 1, 0, 0,
410 31, 6, 0, 1, 14, 41, 1, 10, 7, 0, 0,
420 31, 7, 0, 6, 15, 0, 1, 2, 7, 0, 0]
430 m_vset(4,v)
440 /* AF OM WF SY SP PMD AMD PMS AMS PAN FM SYNTHE
450 v=[44, 15, 2, 1,200,127, 0, 0, 0, 3, 0,
460 /* AR DR SR RR SL OL KS ML DTI DT2 AME
470 31, 0, 0, 0, 0, 22, 0, 2, 3, 0, 0,
480 27, 9, 0, 5, 1, 0, 0, 8, 3, 0, 0,
490 31, 0, 0, 0, 23, 0, 4, 7, 0, 0,
500 27, 9, 0, 6, 1, 0, 0, 4, 7, 0, 0]
510 m_vset(5,v)
520 str a[256],b[256],c[256],d[256],e[256],f[256],g[256],h[256]
530 str j[256],k[256],l[256],m[256],n[256],o[256],p[256],q[256]
540 str s[256],t[256],u[256],v[256],w[256],x[256],y[256],z[256]
550 str a1[256],b1[256],c1[256],d1[256],e1[256],f1[256],g1[256]
560 str a2[256],b2[256],c2[256],d2[256],e2[256],f2[256],g2[256]
570 str j1[256],k1[256],l1[256],m1[256],n1[256],o1[256],q1[256]
580 str j2[256],k2[256],l2[256],m2[256],n2[256],o2[256],q2[256]
590 str s1[256],u1[256],v1[256],w1[256],x1[256],y1[256],z1[256]
600 str s2[256],u2[256],v2[256],w2[256],x2[256],y2[256],z2[256]
610 str u3[256],u4[256],a0[256],b3[256],h3[256],in[256],se1[256]
620 dim str Q(4)[256],R(4)[256],S(4)[256],T(4)[256],U(4)[256],
V(4)[256]
630 sel=bnd("e",12,8192,16383)
640 for i=1 to 14:m_alloc(1,6000):m_assign(i,i):m_trk(1,"@99")
650 m_alloc(11,12000):m_assign(11,11)
660 key 2,"m_play()@M":key 12,"M_STOP()@M":key 19,"SAVE"+chr$(
34)*"RUN"+chr$(5):key 7,"m_trk(1)"
670 m_tempo(182)
680 /* * * * N O T I C E * * *
690 /*EDIT PROG.
700 /* プログラム中で使われている全ての音色の
710 /* JOY STICK というパラメータを
720 /* JOYSTICK P+02 -> P+12
730 /* の様に変更して下さい。
740 /*SEQUENCER
750 /* MIDI CH 1,11,12,13,14,15,16,10
760 /* PAN Tr=9:1 Tr=1:9 Tr=8:2 (Other Tr=5:5)
770 /*GLOBAL (DRUM KIT)
780 /* DRUM KIT 3の"TON 2"を全て"TON 1"にする。
790 /* OPEN HH and CLOSED HHのPANを共に9:1にする。
800 /*
810 /* TRACK 1 (BASS)
820 /*
830 a0="@n1 @47 o2 q8 @m0 @b8192 @d0"
840 in="v12L8r4 @u120b-4f4(e-drc)>b-r4-a-4. r4agg-f4.& f1
850 a="v12@u120L8l:b<b-a-gfgr-e-4(e-d-c)>b<cr>>b-4<b-a-gfgr-e-4
(e-d-c)>d<@u127d>|f@u120e<@u127e>:>|@u120e-16<@u127e>)>
860 x=bnd("e-",12,8192,0)
870 b="@u120L8l:b<b-a-gfgr-e-4(e-d-c)>b<cr>>b-4<b-a-gfgr-e-4
d-16c16>b-4<ce-&@L2">x+@b8192L8>b-&|re-4<e-d-cd-e-r4>
880 bl="b<b-a-gfgr-e-4(e-d-c)>d<@u127d>@u120e<@u127e> t165
v14 r4a-2. t174(gge-e-)t182
890 c="@u120L8l:b<b-a-gfgr-e-4(e-d-c)>b<cr>>b-4<b-a-gfgr-e-4
1c-d-c>d<@u127d>@u120e<@u127e>:>|@u120q4<e>@u120q8d<@u127q
d>@u120q8d<@u127q6d>@u120q8c<@u127q6c>
900 d="L8o2q8a-a<a>a-g-f-e-f-g-4a-a-g-f-d>fe->a-4<a-g-f-e-f>
a-4<a>a-g-f-q6f>q8g<q6g> q8a-a-a-g-g-f-e-f-g-4a-g->f<f<d>->fe-
a-4<a-g-f-e-f>a-4<a>a-g-f<q6f>q8g<q6g>
910 e="L8o2q8g<gcfrcfg>g<gcf>b<b-c<c>>g<gcfrcfg>g<gcfb-afc
>g<gcfrcfg>g<gcf>b<b-c<c>>g<gcfrcfg>g<gcfb-afc
920 g="L8o2q8b-b<b-a-ga-b-r>b-4<b-a-ga-b-e>a->a<a>g<g>e<e-
>rf b<b-a>a<a>g<g>e<e- >b<b-a-ga-b-r>b-4<b-a-ga-b-e>a->a
<a>g<g>e<e-b>b-a<a>g<g>f<fe-e-
930 j="L8o2q8a-a<a>a<a>g<g>a-4a<a>g-a-e-fd-e- >b<q6b>q8g<
q6g>q8a<q6a>q8a<q6a> q8b<b>g<g>a<a>f<f d<d>c<c>>b<b-a>
<a>f<f>g<g>a<a>b<b> daa<d4>dad g<d>g<g>g<dr
940 k="L8v13o2q8l:r1l:r2..b-4<b-a>a-q7gfa-q8b-
950 l="L8l:3 (rb-rb-1l:1 r4<b-16a-16ga-b-r4) |:3 (rb-rb-1l:1
rb-a-ga-gfg
960 m="L8v13o1u120o2q8b-b-a-gf2gq6e-4<q2e-d-c>q7b<cr>>q8b-4
<b-a-gf2gq6e-4<q2e-d-c>q7d<d-16>d-16e<e-16>e-16 b-4<b-ra-gf
>f4<fe-de-f4>b-4<b-a-gr<d-e-4> q8f<q6f>q8e<q6e>q8e<q6e>q8d<q6d
>
970 n=">b<b>a-16<a-16ga-frg e->e<d>->a<a>rg->> b-16<b-16
b-a>b<g>b<b>f<g b<e-a>'fb'@u90'f16b-'g16<c>'@u120'f4b-'>b-
<b> b<b>b-16b-16<a-4ge-f e->e<e>e-16e-16<c>>a<a>gg- f<b-fa-4
gfb<b< d<q6d>q8c<q6c>q8b<q6b>q8a<q6a>
980 o="@n1 @49 o4 q8 v1 @u127 @m0 @b8192L8 @d0r1 b-4v2b-4v3b-4
v5b-4v6b-4v8b-4v10b-4v12b-4 b-1& @m127 b-1& @m0b-1& b-2&b-1b-4
&v5b-4v7b-4 t166v5b-4v3b-4v2b-4v1b-r2 t176@47r1t182
990 x="r>gr2. r1 b-4fr<e-drc4>gr<ferd
1000 m_trk(1,a0+in)
1010 m_trk(1,a)
1020 m_trk(1,b):m_trk(1,b1)
1030 m_trk(1,"v12"+c)
1040 m_trk(1,d)
1050 m_trk(1,e)
1060 m_trk(1,d)
1070 m_trk(1,g)
1080 m_trk(1,j)
1090 m_trk(1,j)
1100 m_trk(1,k)
1110 m_trk(1,l)
1120 m_trk(1,m)
1130 m_trk(1,n)
1140 m_trk(1,g)
1150 m_trk(1,g)
1160 m_trk(1,j)
1170 m_trk(1,o)
1180 m_trk(1,"v13"+c)
1190 m_trk(1,d)
1200 m_trk(1,e)
1210 m_trk(1,d)
1220 m_trk(1,g)
1230 m_trk(1,g)
1240 m_trk(1,"v12"+j)
1250 m_trk(1,x)
1260 /*
1270 /* TRACK 2 (BRASS)
1280 /*
1290 a0="@n12 @3 q8 v7 @m0 @b8192 @d0"
1300 in="@u120L8o5r4 'c4fb-'>'cfb-'r'gb<d>'>'gb<c>'r'gb<c>' r
'gb<d>'r4.'g-lb<ce>'@L72@u127'a<ce-f>' r1@d0
1310 as="@u10L8o5l:'c1fb-' r4.'ce-a-r'ce-a-r'ce-a-' c1fb-' r
l:1
1320 bl="c1fb-'r4'r'ce-a-r'ce-a-r'ce-a-' r4@L144'cfb-'> L1@n
11@3v09@u120'egb<d>'>'egb<c>'>'egb<c>'>'egb<d>'
1330 c="@n11 @3 q8 v08 @m0 @b8192 @u110o5L8r1 r1 r1 r2r@L72'gb-
<e>' r1 r1 L8r2..@u127@u99'e-a-'>'b<e-'>'gb<e-'>'@L72'a<d<->'e-
g'
1340 d="@u120@u99o4q7L8r1 r1 r2r'a<d-f>'r4 r'a<d-g>'r4'a<d-f'
r4. r1 r1 r2..@u127q8g- d>a<f4d>a-a-b<d-e>4
1350 e="@u120@u99o6L8 r1 r2.q4'c4dfg' r2..>'cfb-'>'q6'cfb-' q4r2.'e
4g<c>' r1 r2r'>'c4dfg' r2..@u127>q8g f4g4g16c16f16g16c
1360 g="@76 q8 o5 v13 @u120 @d0 r1 r1 r2..L16a-f d1 r1 r1 r2
b-g-e-c-b-g-e-c&
1370 h="c1 r1 r1 @22o5'a-4<ce-f>r2. r1 r1 r1 r1
1380 j="@n11@3q6o4v0@u105@d0|L8'a<ce-g>'r'a<cdf>'r'a<ce-g>
'r'a<cdf>'r 'a<ce-g>'>'a<ce-a>'>'r'a<cdf>'>'@u127a<cdf>'@u127a<cdf>'@u127a<cdf>'>'r'a<ce-g>'>'q7'fb<e>'>'fb<e>'>'r@L72'fb<cd>' L2'ce-a-b-'>'cdfb
-'>|>'a1<cdg>'L144q8'dfa-b'>|r4
1390 k="@n12 @3 q8 v07"+a
1400 m="|:5r1|@n11o5q4@u75L8r2d-e-4. r1 r1
1410 n="|:8r1|
1420 o="@n12 @3 q8 v08 @u110 @d0L8o5'c1fb-' r4.'ce-a-r'ce-a-r'
'ce-a-' c1fb-' r1+bl
1430 r="@u120@u99o4q7L8r1 r1 r2r'a<d-f>'r4 r'a<d-g>'r4'a<d-f'
r4. r1 r1 r2..q8g- d>a<f4d>a-a-b<d-e>4
1440 w="@n11@3q6o4v0@u105@d0|L8'a<ce-g>'r'a<cdf>'r'a<ce-g>
'r'a<cdf>'r 'a<ce-g>'>'a<ce-a>'>'r'a<cdf>'>'@u127a<cdf>'@u127a<cdf>'@u127a<cdf>'>'r'a<ce-g>'>'q7'fb<e>'>'fb<e>'>'r@L72'fb<cd>' L2'ce-a-b-'>'cdfb
-'>|>'a1<cdg>'L8l:4r'dfa-b'>|
1450 x="q8r'dfa-b'r2. r1 'c4fb->'L8'cfb-'r'gb<d>'>'gb<c>'>'r@L7
2'dg<c>'L8>'dg<c>'>'ga<ce>'>'f+a<cd>'>'r'dfa<c>'
1460 m_trk(2,a0+in)
1470 m_trk(2,a)
1480 m_trk(2,a):m_trk(2,b1)
1490 m_trk(2,c)
1500 m_trk(2,d)
1510 m_trk(2,e)
1520 m_trk(2,d)
1530 m_trk(2,g)
1540 m_trk(2,h)
1550 m_trk(2,j)
1560 m_trk(2,k)
1570 m_trk(2,a)
1580 m_trk(2,m)
1590 m_trk(2,n)
1600 m_trk(2,g)
1610 m_trk(2,h)
1620 m_trk(2,j)
1630 m_trk(2,o)
1640 m_trk(2,c)
1650 m_trk(2,r)
1660 m_trk(2,e)
1670 m_trk(2,r)

```



```

1680 m_trk(2,g)
1690 m_trk(2,h)
1700 m_trk(2,w)
1710 m_trk(2,x)
1720 /*
1730 /* TRACK 3 (PIANO)
1740 /*
1750 a0="en13 @72 o5 q8 v09 @m0 @b8192 @d0"
1760 in="e09L8o5r4 'c4fb-r4'gb-<d>'gb-<c>'r'gb-<c>' r'gb-<d>'
>r4 'b-1ce->'@L72@127'a-<ce-f>' r1@d0
1770 a="e015L8o5l:'c1fb-r4.'ce-a-'r'ce-a-'r'ce-a-' 'c1fb-r4
1780 b1="c1fb-r4'r'ce-a-'r'ce-a-'r'ce-a-' 'r4@L144'cfb->' L4@n
12@52v07q6'egb-<d>'egb-<c>'egb-<c>'egb-<d>'
1790 c="L8l:'cfb-r'r'ce-a-'r'r'b-<e-a-'r' l1r'e-gb-'r'r'e-a-<c>'r
4: l1 r'b-<e-g'r4.'>g4b-<e-'r
1800 c1="cfb-r'r'ce-a-'r'r'b-<e-a-'r' r'e-gb-'r'r'e-a-<c>'r4. 'c
fb-r'r'ce-a-'r4.@d127a- e->b-<d-4.g4.
1810 d="e0d0'r'b-<e-a-'r'r'b-<e-g'r4'>a-<d-f'r' r'a-a-4<d-g-'r'd-g-
b'r4. 'b-<e-a-'r'r'b-<e-g'r'>a-<d-f'r4 r'a-<d-g-'r4'>a-<d-f'r4
1820 d1="r'a-<e-a-'r'r'b-<e-g'r4'>a-<d-f'r' r'a-a-4<d-g-'r'd-g-b-
'r4. 'a-<e-a-'r'r'b-<e-g'r4.@d127g- d>a-<f4d>a-<a-b-<d-e-j4
1830 e="e0d0l:'>g<cf'r'r'>g<ce'r4'cfb-r' l1r'c2fa'cdf'gr:l1 r'c4f
a'r4. 'e-g<c>'r
1840 e1="g<cf'r'r'>g<ce'r4'cfb-r' r'c2fa'cdf'gr'>g<cf'r'r'>g<
ce'r4.@d127g fc4g4g16c16f16g16c>
1850 g="l:'e0d0L72q8'cdfb-'@d127'a-1b-<df'L8r'l1@d0'fa-b-<e->'f
fa-b-<d>'@d127'e-1fa-<c>'r'r1 :l@d0'fa-b-<e->'fa-b-<f>'r@L168'e
gb-'L8r@L127'e-1gb-<c>'r
1860 h="l:'e0d0L72q8'cdfb-'@d127'a-1b-<df'L8r'l1@d0'fa-b-<e->'f
a-b-<d>'@d127'e-1fa-<c>'r'r1 :l@d0'fa-b-<e->'fa-b-<f>'r@L168'e
gb-'r@L144'e-gb-<c>'L8'a-<ce-g>'r'a-<ce-g>'
1870 j="l:'L8'a-<ce-g>'r'a-<cdf>'r'a-<ce-g>'r'a-<cdf>'r' 'a-<ce-g
>'r'a-<ce-a>'r'a-<cdf>'r@L127a-<c'df'@d0>' l1'a-4<ce-g>'q7'fb-<ce
->'fb-<ce->'r@L72'fb-<cd>'L2'ce-a-b-'cdfb-' :l1 'a1<cdg>'dfa-
b'
1880 k="q8 v09o5"+a
1890 m="en11 @05 o4 q6 v08 @u10L8r'cdfb-'r4'cdfb-'r4@L72'ce-fa
-L8'ce-fa-'r'ce-g'ce-fa-'r4 r'a-<ce-g'rql'>a-<ce-q4'>a-<ce-g
'q1'c16e-fa-'>a-16<ce-fa-'q4'>a-<ce-g'q1'ce-fa-' q5'c4e-fa-'r@d
127L52dfgb-r2
1900 m1="e0d0L8q6r'cdfb-'r4'cdfb-'r4@L72'ce-fa-'L8'ce-fa-'r'ce-g
'@L72'ce-fa-' L8r'a-<ce-g'r4'>a-<ce-g'r'ce-fa-'ce-fa-' @u90q2'
>b-4<dfb-'q1'>'f16b-<df'>'f16<f'>'f4b-<df'>'>b-<dfa-'q8'>'b-4<dfgb-
'@u110
1910 n="q6'r'cdfb-'r4q6'c4e-fa-'q2'ce-fa-'ce-fa-' q4'a-4<ce-g>'
'c4e-fa-'q4'cdfb-'r4q4'c4d32e-gb-'r32'd32b-'r32q4'de-gb-'de-gb-
-'r'de-gb-'r'de-gb-' q6'c4e-fa-'r8q8@L127L32cdfb-r2
1920 n1="e0d0L8q6r'c4dfb-'r4'cdfb-'r4@L72'ce-fa-'L8'ce-fa-'r'ce
-g'q5@L72'ce-fa-' L4q7'ce-fb-'ce-g'ce-fa-'ce-fa-' L8@L127'b->b
-<'f>'>'e->e-<'d2>d<'>' /r8
1930 o="q8 o5 @u115 @d0 L8'clfb-r4.'ce-a-'r'ce-a-'r'ce-a-' 'c
1fb-r4.'b1
1940 r1="r'a-<e-a-'r'r'b-<e-g'r4'>a-<d-f'r' r'a-a-4<d-g-'r'd-g-b-
'r4. 'a-<e-a-'r'r'b-<e-g'r4.g- d>a-<f4d>a-<a-b-<d-e-j3
1950 u="l:'L8'a-<ce-g>'r'a-<cdf>'r'a-<ce-g>'r'a-<cdf>'r' 'a-<ce-g
>'r'a-<ce-a>'r'a-<cdf>'r@L127a-<c'df'@d0>' l1'a-4<ce-g>'q7'fb-<ce
->'fb-<ce->'r@L72'fb-<cd>' L2'ce-a-b-'cdfb-' :l1 'a1<cdg>'L8l:4r
'dfa-b' :l
1960 m_trk(3,a0+in)
1970 m_trk(3,a)
1980 m_trk(3,a):m_trk(3,b1)
1990 m_trk(3,"en12 o5 q7 v07 @m0 @b8192 @u115"+c):m_trk(3,c
1)
2000 m_trk(3,d):m_trk(3,d1)
2010 m_trk(3,e):m_trk(3,e1)
2020 m_trk(3,d):m_trk(3,d1)
2030 m_trk(3,g)
2040 m_trk(3,h)
2050 m_trk(3,j)
2060 m_trk(3,"en13 @72"+k)
2070 m_trk(3,a)
2080 m_trk(3,m):m_trk(3,m1)
2090 m_trk(3,n):m_trk(3,n1)
2100 m_trk(3,"en12 @52 o6 q7 v07 @b8192 @u115 r8"+g)
2110 m_trk(3,h)
2120 m_trk(3,j)
2130 m_trk(3,"v09"+o)
2140 m_trk(3,"en12 o6 q7 v7 @m0 @b8192 @u115"+c):m_trk(3,c1)
2150 m_trk(3,d):m_trk(3,r1)
2160 m_trk(3,e):m_trk(3,e1)
2170 m_trk(3,d):m_trk(3,r1)
2180 m_trk(3,g)
2190 m_trk(3,h)
2200 m_trk(3,w)
2210 m_trk(3,x)
2220 /* MT-32
2230 m_trk(6,"en2 @4 o5 q7 v06 @m0 @b8292 @u90 p1"+in)
2240 m_trk(6,"f:20r1:l")
2250 m_trk(6,"o5"+c):m_trk(6,c1)
2260 m_trk(6,d):m_trk(6,d1)
2270 m_trk(6,e):m_trk(6,e1)
2280 m_trk(6,d):m_trk(6,d1)
2290 m_trk(6,g)
2300 m_trk(6,h)
2310 m_trk(6,"p2"+j)
2320 m_trk(6,"v6 o5 @p10"+a)
2330 m_trk(6,a)
2340 m_trk(6,"l:16r1:l")
2350 m_trk(6,"o5 q7 v06 @b8292 @u90 p1"+g)
2360 m_trk(6,h)
2370 m_trk(6,"p2"+j+p1)
2380 m_trk(6,o)
2390 m_trk(6,"en2 o5 q7 v06 @b8292 @u90 p1"+c):m_trk(6,c1)
2400 m_trk(6,d):m_trk(6,d1)
2410 m_trk(6,e):m_trk(6,e1)
2420 m_trk(6,d):m_trk(6,d1)
2430 m_trk(6,"o5 q7 v06 @b8292 @u90 p1"+g)
2440 m_trk(6,h)
2450 m_trk(6,"p2"+u+p1)

```

```

2460 m_trk(6,x)
2470 /*
2480 /* TRACK 4 (PIANO 2)
2490 /*
2500 a0="en14 @3 o5 q8 v07 @m0 @d0
2510 in="r4 'l:r1:l"
2520 a="b8292@u10L8l:'c1fb-r4.'ce-a-'r'ce-a-'r'ce-a-' 'c1fb
-r' r1 :l
2530 b1="c1fb-r4'r'ce-a-'r'ce-a-'r'ce-a-' 'r4@L144'cfb->' L4@n
13@72v08q6'egb-<d>'egb-<c>'egb-<c>'egb-<d>'
2540 c="L8l:'cfb-r'r'ce-a-'r'r'b-<e-a-'r' l1r'e-gb-'r'r'e-a-<c>'
r4 :l r4'>b-4<e-g'r4'>g4b-<e-'
2550 c1="r'cfb-r4'ce-a-'r'r'b-<e-a-'r' r4'e-gb-'r'r'e-a-<c>'r4 r'
cfb-r4'ce-a-'r4@L127a- e->b-<d-4.g4.
2560 d="e0d0'r'b-<e-a-'r'r'b-<e-g'r4'>a-<d-f'r' r4'a-4<d-g-'r'd-g-
b'r4 r'b-<e-a-'r'r'b-<e-g'r4'>a-<d-f'r4'>a-<d-g-'r4'>a-<d-f'r4
4
2570 d1="r'a-<e-a-'r'r'b-<e-g'r4'>a-<d-f'r' r4'a-4<d-g-'r'd-g-b-
-r4 r'a-<e-a-'r'r'b-<e-g'r4@L127g- d>a-<f4d>a-<a-b-<d-e-j4
2580 e="e0d0l:r'>g<cf'r4'>g<ce'r'r'cfb-' l1r4'c4fa'r'cdf'gr:l1 r4
c4fa'r4. 'e-g<c>'r
2590 e1="r'g<cf'r4'>g<ce'r'r'cfb-' r4'c4fa'r'cdf'gr r'>g<cf'r4'
>g<ce'r4@L127g fc4g4g16c16f16g16c>
2600 g="b8292l:'e0d0L72q8'cdfb-'@d127'a-1b-<df'L8r'l1@d0'fa-b-<
e->'fa-b-<d>'@d127'e-1fa-<c>'r'r1 :l@d0'fa-b-<e->'fa-b-<f>'r@L
168'e-gb-'L8r@L127'e-1gb-<c>'r
2610 h="l:'e0d0L72q8'cdfb-'@d127'a-1b-<df'L8r'l1@d0'fa-b-<e->'f
a-b-<d>'@d127'e-1fa-<c>'r@L168'e-gb-'r@L144'e-gb-<c>'L8'a-<ce-g>'
r'a-<ce-g>'
2620 j="e0b8192l:8 r1 :l
2630 k="q8 v07o5"+a
2640 m="en12 @05 o4 q6 v07 @u90 @b8292L8r32r'cdfb-'r4'cdfb-'r4@
L72'ce-fa-'L8'ce-fa-'r'ce-g'ce-fa-'r4 r'a-<ce-g'rql'>a-<ce-q4'>a-<ce-g
'q1'c16e-fa-'>a-16<ce-fa-'q4'>a-<ce-g'q1'ce-fa-' q6'c4
e-fa-'r@d127L32dfgb-r2
2650 m1="e0d0L8q6r'cdfb-'r4'cdfb-'r4@L72'ce-fa-'L8'ce-fa-'r'ce-g
'@L72'ce-fa-' L8r'a-<ce-g'r4'>a-<ce-g'r'ce-fa-'ce-fa-' @u70q2'
>b-4<dfb-'q1'>'f16b-<df'>'f16<f'>'f4b-<df'>'>b-<dfa-'q8'>'b-4<dfgb-
'@u90
2660 o="en13 @3 q8 v07 o5@b8292@u10@d0L8'clfb-r4.'ce-a-'r'ce
-a-'r'ce-a-' 'c1fb-r4.'b1
2670 r1="r'a-<e-a-'r'r'b-<e-g'r4'>a-<d-f'r' r4'a-4<d-g-'r'd-g-b-
-r4 r'a-<e-a-'r'r'b-<e-g'r4.g- d>a-<f4d>a-<a-b-<d-e-j4
2680 x="q8r1 r1 'c1fb-r4'cfb-r'r'gb-<d>'gb-<c>'r@L72'gk<c>'
L8'>dg<c>'r'ga<ce>'f+a<cd>'r'dfa<c>'
2690 m_trk(4,a0+in+a)
2700 m_trk(4,a):m_trk(4,b1)
2710 m_trk(4,"en13 o5 q7 v08 @m0 @b8192 @u115"+c):m_trk(4,c
1)
2720 m_trk(4,d):m_trk(4,d1)
2730 m_trk(4,e):m_trk(4,e1)
2740 m_trk(4,d):m_trk(4,d1)
2750 m_trk(4,g)
2760 m_trk(4,h)
2770 m_trk(4,j)
2780 m_trk(4,"en14 @3"+k)
2790 m_trk(4,a)
2800 m_trk(4,m):m_trk(4,m1)
2810 m_trk(4,n):m_trk(4,n1)
2820 m_trk(4,"en13 @72 o5 q7 v08 @u115 r16"+g)
2830 m_trk(4,h)
2840 m_trk(4,j)
2850 m_trk(4,o)
2860 m_trk(4,"en13 @72 o5 q7 v08 @m0 @b8192 @u115"+c):m_trk(4,c
1)
2870 m_trk(4,d):m_trk(4,r1)
2880 m_trk(4,e):m_trk(4,e1)
2890 m_trk(4,d):m_trk(4,r1)
2900 m_trk(4,g)
2910 m_trk(4,h)
2920 m_trk(4,j)
2930 m_trk(4,j)
2940 /* MT-32
2950 m_trk(7,in+l:20r1:l")
2960 m_trk(7,"en3 @l o4 q7 v08 @m0 @b8292 @u99 p2"+c):m_trk(7,c
1)
2970 m_trk(7,d):m_trk(7,d1)
2980 m_trk(7,e):m_trk(7,e1)
2990 m_trk(7,d):m_trk(7,d1)
3000 m_trk(7,g)
3010 m_trk(7,h)
3020 m_trk(7,j)
3030 m_trk(7,"l:32r1:l")
3040 m_trk(7,"o4 q7 @b8292 @u99 p2"+g)
3050 m_trk(7,h)
3060 m_trk(7,j)
3070 m_trk(7,"l:8r1:l")
3080 m_trk(7,"en3 o4 q7 v08 @b8292 @u99 p2"+c):m_trk(7,c1)
3090 m_trk(7,d):m_trk(7,d1)
3100 m_trk(7,e):m_trk(7,e1)
3110 m_trk(7,d):m_trk(7,d1)
3120 m_trk(7,g)
3130 m_trk(7,h)
3140 m_trk(7,j)
3150 m_trk(7,x)
3160 /*
3170 /* TRACK 5 (BRASS 2)
3180 /*
3190 a0="en15 @73 q8 v11 @m0 @b8292 @d0"
3200 in="e0120L8o6r4 'c4fb-r'cfb-r'gb-<d>'gb-<c>'r'gb-<c>' r
'gb-<d>'r4.'g-1b-<ce->'@L72@127'a-<ce-f>' r1@d0
3210 a="e010L8o6l:'c1fb-r4.'ce-a-'r'ce-a-'r'ce-a-' 'c1fb-r4
1 :l
3220 b1="c1fb-r4'r'ce-a-'r'ce-a-'r'ce-a-' 'r4@L144'cfb->' r1
3230 c="en15 @73 q8 v11 @m0 @b8292 @u110o6L8r1 r1 r1 r2r@L72'gb
-<e->' r1 r1 L8r2.@d127@u99'e-a-' 'b-<e-'>'gb-<@L72'>a-<d-'>'e-
g'
3240 d="e0d0@u99o5q7L8r1 r1 r2r'a-<d-f>'r4 r'a-<d-g>'r4'a-<d-f>'
r4. r1 r1 r2.@d127q8g- d>a-<f4d>a-<a-b-<d-e-j4
3250 e="e0d0@u99o5q7L8r1 r1 r2.q4'c4dfg' r2.'cfb-<'q6'cfb-' q4r2.'c
4e-g' r1 r2r8'cdfg'c4dfg' r2..@d127>q8g fc4g4g16c16f16g16c>

```


●リンカWLK

今月はリロケータブルファイルを実行可能なマシン語ファイルに変換するリンカWLKの登場です。マシン語プログラムを開発する場合には先月発表したWZDと今月のWLKの2つが必要ですので注意してください。WZDとWLKはいわば表裏一体で、どちらが欠けても用をなしません。

それならば、なぜまとめてひとつのプログラムにしてしまわないのかと疑問を持たれるかもしれませんね。リロケータブルファイルはマシン語ファイルの一種ですが、CALL命令やJP命令の飛び先などのアドレスがまだ確定していない不完全なマシン語ファイルです。

逆にリロケータブルファイルのこの特性を利用すると、CALL先がファイル内になくてもアセンブルできるというメリットが生まれます。これらの不完全なマシン語ファイルをいくつかつなぎ合わせ、未確定のアドレスなどを確定して完全なマシン語ファイルにするのがリンカの役割です。

リンカの役割はリロケータブルファイルを実行可能なマシン語ファイルにすることだけであり、そのリロケータブルファイルがどんな処理系によって出力されようと思ったことではないのです。

第97部

リンカWLK

リロケータブルファイルを出力するのはCコンパイラかもしれないし、PASCALコンパイラかもしれない。JP先やCALL先がプログラム内になくてもいいのですから、メインとなる部分はC言語を使って書き、速度を要求される部分や緻密な処理を行いたい部分だけをアセンブリ言語で書くという技も使えます。リロケータブルファイルを扱う世界ではシステムプログラムはリロケータブルファイルを出力するものとそれを実行可能なマシン語ファイルに交換するものの2つに大きく分けることができるといえるでしょう。

WZDは前者に当たります。数ある（今後登場するかもしれない）リロケータブルファイルを出力するもののひとつにすぎません。そしてWLKは後者です。表裏一体となって使われる2つのプログラムがひとつにまとめられていない理由がおわかりいただけたでしょうか。

●四方山話

編集室に面白いゲームが届きました。編集者とライターの間で対戦が盛り上がっています。一世を風靡したこのゲームを来月はお届けすることにし、ライブラリアンは再来月ということになりました。ご期待ください。



●S-OSの系譜 (12)

S-OS “SWORD” はCP/MやMS-DOSなどのDOSと比較して足りない機能がいくつか存在していました。ひとつは何度か書いたようにファイルの扱いが弱いということです。ファイルとの1文字単位の入出力はサポートされていませんし、ファイル内をシークするなんてこともできません。しかし、こういった機能をほしい人（つまりファイル処理を行うS-OSのアプリケーションを自分で作成しようとする人）なら、用意されている機能を使って好きなように実現できる機能ですから、声高に叫ぶほどのことではないでしょう。

それより、掲載されたプログラムを入力して使う人にとってはユーザーフレンドリでないという点のほうが問題だったのではないのでしょうか。CP/MやMS-DOSがユーザーフレンドリだと思う人はいないでしょうが、S-OSのユーザーインタフェースはそれに輪を掛けて冷たいものでした。

そもそもユーザーインタフェースなどと呼べるシェルは存在せず、モニタという名のコマンド解釈プログラムが用意されているだけだったのだから当然です。プログラムを実行するには、まずマシン語ファイルをロードし、実行開始番地へジャンプする。まさにマシン語モニタ感覚です。

1986年10月号では、このモニタを機能拡張しようという試みが行われました。まず、マシン語ファイルのロード・実行を一気に行うRコマンドをモニタに追加。さらに、常用できる小さなプログラムとして、ファイルのコピープログラム（S-OSではASCIIファイルのコピーができない）とファイルの内容を画面に表示するプログラムが発売されました。これにより、モニタの使い勝手も随分向上しました。この試みは1987年5月号のS-OS “SWORD” 変身セットへと受け継がれていくことになります。

同時に発表されたディスクモニタDREAMは、ディスク上のファイルを連続クラスタに収め直す、ディレクトリを並べ替えるといった便利な機能を備えたツールとして人気を集めました。

リンカWLK

Ishigami Tatsuya

石上 達也

お待ちせしました。ついにリンカWLKの登場です。これでWZDでアセンブルしたオブジェクトファイルを実際に起動できるファイルにすることができます。これらWZD、WLKはS-OSに新しい開発環境を築いてくれることでしょう。

WLKとは?

皆さん先月号のWZDは、もう入力し終わったでしょうか? なにせ9Kバイトにも及ぶ超大作だったので大変だったでしょう。入力するのも大変ならデバッグするのは、その10倍大変なのです(システム関係は、ゲームと違いデバッグ=気分転換にならない)。できるだけ多くの人に使っていたきたいものです。

さて、その9Kバイトにも及ぶWZDとともに大規模アプリケーションの開発に威力を発揮するのが、このプログラムです。ネーミングはWZD専用のリンカということ、WLKとしました。とくにアルファベットの語呂合せは、考えていません。今回は約7Kバイトの分量です。がんばって入力してください。

リンカとは?

先月号でも少しお話ししましたが大規模なプログラム開発には、リロケータブルアセンブラを用いると有利です。そのとき、リロケータブルアセンブラから出力された個々のリロケータブルファイルをつなげ(リンクし)1本のオブジェクトファイルを出力するのがこのリンカです。

ざっと使い方を説明しましょう。プロンプト'*'が出ている状態が、入力可能状態です。ここでは、WZDのリンク作業を例にとって説明します。

まずWZDのメモリ配置は、3000H番地からコードセグメント、6000H番地からがデータセグメントですので、それぞれのスタートアドレスをリンカに知らせてやりませう。それには、

*P:3000,/D:6000

と入力します。そして、WZD本体は、WZD1、WZD2、WZD3、WZD35、WZD4から成っていますので、これらをリンクします。

*WZD1

ここまで入力し終わったなら、画面上には、なにやら表が表示されるはずです。これは、WZD1をリンクした時点での、未定義ラベルとそのラベルが初めて使用された場所です。次に、

*WZD2,WZD3,WZD35,WZD4

と、入力してください。今度はなにも表示されずに、プロンプト'*'が表示されたはずです。なぜなら、WZDのすべてのファイルをリンクし終えたので、未定義のラベルはもうないからです。

この状態で、もしラベルリストがほしいければ、

*WZD/M

と入力してください。ファイル“WZD.MAP”にラベルリストが収録されています。

次に、WZDはデータセグメントに初期条件などを置いていないので(つまり、実際に必要なのはコードセグメントのみなので、

*WZD/N:P

と入力すると、“WZD.OBJ”というファイル名で、オブジェクトファイルを作ります。

*WZD/N

と、最後にやってしまうとデータセグメントに割り振ったワークエリア領域までも含んだファイルができてしまいます。これでも一応は動くWZD.OBJができるのですが、かなりの無駄な部分まで含んでしまいます。

と、こんな具合にリンク作業は行いますが、要は、/P:スイッチと/D:スイッチで、スタートアドレスを決めてやり、リンクしたいファイルの名前を打ち込んであげればいいのです。そうしたら、Nスイッチ

で、それらをオブジェクトファイルとして、取り出せます。

プログラム

このプログラムもWZDと同様に腕力にものをいわせて作ったものです。それぞれのアイテムに応じて、適当な処理を行います。サブルーチンも何カ所か同じようなものが出てきたら、適宜作っていくという感じですが、WZDと違いリンカは、これでも作れた。

サブルーチンごとに、ほとんどが独立しているもので難所はそんなに困難ではないと思います。

WZDと同様にWLK1.ASMは、最初S-OSのソースコードを用いて作成しました。このとき、
1) コーカル変数は、だいたい1関数につき5つ以内(BC、DE、HLレジスタに対応させることができる)
2) マークされたポインタは2つ以内(IX、DXレジスタに対応させることができる)
という条件が満たされていたのでハンドコーディングは、たいへんスムーズに行えました。

先になるソースですが(プログラムのサイズが、念のため)、だいたいSmall-Cでコンパイルしたときの1/3程度になりました(ライブラリは除いて。本体のみ)。

メモリーマップの変更は、リンカに与えるパラメータとWLKのヘッダファイルであるWZD.Hの内容を変更することによって行えます。現在、コーカルなラベルは1024個変更できるようになっていますが、この数を変更するときもこのWLK.Hを書き換えます。

WLKをアースからアセンブルするとき

お詫びと訂正

実は、先月号の私の記事に少しばかり誤りがありましたので、ここに訂正させていただきます。まず、プログラムリストの呼び方が、記事中与注記中では、異なっていたこと。注記中のリスト2から6までは、それぞれ順に、

WZD1,WZD2,WZD3,WZD35,WZD4
です。

また、79ページの中段31行目を、

表1

■起動方法

“SWORD”の拡張をしていない人はコマンドラインから、
#LWLK
#J3000
と、拡張をしてある人は、
#WLK
で起動します。すると、この後ろにプロンプト
‘*’を表示してパラメータの入力待ちになります。
なお、それぞれ、
#J3000
#WLK
の後ろにパラメータを書くことができます。

■パラメータ

以下、[ファイル名]とあるのは、大文字小文字を区別し、スイッチ (/Sとか/Nとか)は、どちらでもかまいません。
* [ファイル名]
[ファイル名]で表されたリロケータブルファイルを取り込みます。省略時の拡張子は‘.REL’です。
* [ファイル名] / S
[ファイル名]で表されたライブラリファイル、未定義なモジュールに限り取り込みます。省略

*WZD1,WZD2,WZD3,WZD35,WZD4,
WZD/N:P

としてください。

* * *

6月16日のコンサートにきてくださった皆様ありがとうございます。おかげさまで、コンサートは大成功をおさめることができました。ちなみに、コンサートのプログラムの役員紹介の写真で、コンサート委員長をひっくり返して下から支えているのが私です。来年もやりますので、ぜひ、またきてください。

WZD
*=WLK1
*=WLK2
*=WLK3
* [ここでシフト+ブレイクを押す]
WLK
*/P:3000,/D:4500
*WLK1,WLK2,WLK3,WLK/N:P
とすれば、ここに掲載されているものと同様なオブジェクトが得られます。

いうまでもないと思いますが、ソースリストのみを一生懸命に入力しても、WZDとWLKのオブジェクト形式のプログラムがなければ、アセンブラを通してオブジェクトを得るということはできません。まず最初に、WZDとWLKのオブジェクト・プログラムが絶対に必要です。

最後にばらしてしまいますが、べき乗を行うサブルーチンは、第1パラメータ(Hレジスタ)を、第2パラメータ(DEレジスタ)回掛け合わせるということをしています。本当は、

$$A^B = \text{EXP}(B * \text{Log}(A))$$

を展開して、ゴリゴリ計算したかったのですが、メモリを大量に使ううえ、浮動小数のレベルで計算しなければ精度が出ないので、しかたなしに、中学1年生しています。腕に自信のある方、なにかよい方法をご存じの方はご連絡ください。

WZDの訂正

コマンドラインからのRUNコマンドを拡張した“SWORD”を使用した場合、処理を終了して“SWORD”のモニタに戻ってくるときに誤動作する場合があります。

```
3008 ED 5B 76 1F → CD AB 50 00
303A 2A 76 1F → CD B2 50
3153 ED 7B 00 → C3 FA 1F
347A ED 7B 00 → C3 FA 1F
```

以上のように訂正したうえ、次のダンプリストのようにプログラム(50B7Hまで)を追加してください。

```
5080 3E 09 37 18 04 3E 80 90 : E8
5088 A7 E1 C1 C9 26 00 6F 29 : D0
5090 29 29 29 C9 E5 CB 3C CB : FB
5098 1D CB 3C CB 1D CB 3C CB : DE
50A0 1D CB 3C CB 1D 7D E1 C9 : 33
50A8 00 00 00 ED 5B 76 1F 13 : F0
50B0 13 C9 2A 76 1F 23 23 C9 : AA
50B8 00 00 00 00 00 00 00 00 : 00
50C0 00 00 00 00 00 00 00 00 : 00
50C8 00 00 00 00 00 00 00 00 : 00
50D0 00 00 00 00 00 00 00 00 : 00
50D8 00 00 00 00 00 00 00 00 : 00
50E0 00 00 00 00 00 00 00 00 : 00
50E8 00 00 00 00 00 00 00 00 : 00
50F0 00 00 00 00 00 00 00 00 : 00
50F8 00 00 00 00 00 00 00 00 : 00
-----
SUM: 5B 72 C3 A3 C3 EA 8A F4 5C85
```

表2 エラーメッセージ

Undefined Item-xx

未定義アイテムxxを使用した。つまり指定されたリロケータブルファイルの内容がおかしい。WZDを使用している場合には起こらない。

Multi Defined

同じラベル名が、2カ所以上で定義されている。このメッセージの後ろに16進2桁の数字が表示されたらそのラベル番号を持つ内部ラベルがエラーであり、文字列が表示されたらその名前の外部ラベルがエラーである。

Undefined Label

未定義ラベルが使用された。内部エラーと外部エラーの区別については、Multi Definedエラーのときと同様である。

Too Many Labels

使用されたラベルの数が多すぎる。

Too Far

相対ジャンプ関係のオブジェクトを作成しよ

うとしたが、目的のアドレスが相対ジャンプで届く範囲にない。

Stack Over

演算用のスタックがオーバーした(16レベル以上のスタックが使用された)。

Stack Empty

演算用のスタックが空なのにその内容を参照するアイテムが使用された。

Illegal ORG to xxxx

PC(Position Counter)を後ろ向きにアドレスxxxxに変更しようとした。以下のオブジェクトファイルの内容は保証されない。

DSEG buffer is Over flowed !!

DSEGエリア用のバッファが足りなくなった(DSEG用のバッファは8Kバイト、これ以上のメモリを用いる場合は、WLK.H内のメモリーテーブルを変更するか、ほかのセグメントに割り振ってください)。

リスト1

```

3000 ED 7B 6A 1F ED 5B 76 1F : CE
3008 13 13 06 01 1A 13 A7 28 : 29
3010 07 FE 20 20 F7 04 18 F4 : 4C
3018 C5 CD 23 40 CD 45 40 21 : 68
3020 00 30 22 45 45 22 44 45 : 88
3028 22 42 45 22 58 45 22 56 : E0
3030 45 AF 32 3D 45 32 14 45 : 33
3038 32 28 45 32 3C 45 32 48 : CC
3040 45 32 49 45 DD 21 00 50 : 53
3048 F1 FE 01 20 05 CD 0B 34 : 21
3050 18 11 FD 2A 76 1F FD 23 : 05
3058 FD 23 FD 7E 00 FD 23 FE : B9
3060 20 20 F7 FD 7E 00 A7 20 : 79
3068 06 CD 92 3C CD 0B 34 11 : BE
3070 00 45 FD E5 E1 CD C6 36 : D1
3078 E5 FD E1 21 74 33 CD 77 : CF
SUM: BB 35 3C A3 E1 AA BA 07 5EC0

```

```

3080 36 21 78 33 C4 77 36 20 : 93
3088 34 CD EA 36 E5 ED 5B 56 : A1
3090 45 B7 ED 52 E1 DC 53 36 : 84
3098 22 56 45 DD 36 00 01 DD : AE
30A0 23 DD 75 00 DD 23 DD 74 : C6
30A8 00 DD 23 3A 48 45 A7 C2 : 30
30B0 B7 32 3E 01 32 48 45 22 : 09
30B8 42 45 C3 B7 32 21 7C 33 : 03
30C0 CD 77 36 21 80 33 C4 77 : 89
30C8 36 20 3A CD EA 36 E5 ED : 4F
30D0 58 58 45 B7 ED 52 E1 DC : AB
30D8 53 36 22 58 45 22 5A 45 : 09
30E0 DD 36 00 02 DD 23 DD 75 : 67
30E8 00 DD 23 DD 74 00 DD 23 : 51
30F0 3A 49 45 A7 C2 B7 32 3E : 58
30F8 01 32 49 45 22 44 45 22 : 8E
SUM: B6 DF B5 52 1A 0C 3F 91 AD97

```

```

3100 46 45 C3 B7 32 21 84 33 : 0F
3108 CD 77 36 21 87 33 C4 77 : 90
3110 36 C2 A7 31 DD 36 00 04 : E7
3118 DD 23 21 00 45 CD 89 36 : F2
3120 11 00 45 21 D3 33 DC 93 : EC
3128 36 DD E5 D1 21 00 45 CD : FC
3130 AF 36 E5 DD E1 3E 01 11 : D8
3138 00 45 CD BC 40 DA 19 36 : 37
3140 CD 26 34 DD 22 3E 45 2A : D3
3148 3E 45 7E 23 FE FF CA 98 : 83
3150 31 FE 0E C2 34 36 5E 23 : BC
3158 56 23 DD 53 40 45 11 8D : DC
3160 45 7E 23 12 13 A7 20 F9 : CB
3168 22 3E 45 CD 14 3C D2 47 : DF
3170 31 CA 47 31 2A 40 45 DD : DB
3178 75 00 DD 23 DD 74 00 DD : A3
SUM: BB 0B A8 DC B2 F1 C1 F7 9559

```

```

3180 23 CD 82 42 DA 34 36 2A : 22
3188 B0 45 22 AE 45 3E 02 32 : 7C
3190 53 45 CD 28 37 C3 47 31 : FF
3198 DD 36 00 00 DD 23 DD 36 : 26
31A0 00 00 DD 23 C3 B7 32 21 : CD
31A8 8A 33 CD 77 36 21 8D 33 : 18
31B0 CA 77 36 20 18 11 28 45 : 2A
31B8 21 00 45 CD AF 36 21 28 : 61
31C0 45 CD 89 36 11 28 45 21 : 70
31C8 D7 33 DC 93 36 C3 B7 32 : 5B
31D0 21 90 33 CD 77 36 21 95 : 14
31D8 33 CA 77 36 20 28 DD 36 : 02
31E0 00 FF 11 14 45 21 00 45 : CF
31E8 CD AF 36 21 14 45 CD 89 : 82
31F0 36 11 14 45 21 DB 33 DC : AB
31F8 93 36 3E 01 32 3C 45 32 : ED
SUM: 78 B0 3E E6 80 40 A3 7E EB0A

```

```

3200 60 45 AF 32 61 45 C3 B7 : A6
3208 32 21 9A 33 CD 77 36 21 : B8
3210 90 33 CA 77 36 20 2B DD : 69
3218 36 00 FF 11 14 45 21 00 : C0
3220 45 CD AF 36 21 14 45 CD : 3E
3228 89 36 11 14 45 21 DB 33 : 58
3230 DC 93 36 3E 01 32 3C 45 : 97
3238 32 61 45 AF 32 60 45 C3 : 21
3240 B7 32 21 9A 33 CD 77 36 : 5B
3248 21 A7 33 CA 77 36 20 2A : B6
3250 DD 36 00 FF 11 14 45 21 : 9D
3258 00 45 CD AF 36 21 14 45 : 71
3260 CD 89 36 11 14 45 21 DB : F2
3268 33 DC 93 36 3E 01 32 3C : 85
3270 45 32 61 45 32 60 45 C3 : B7
3278 B7 32 DD 36 00 03 DD 23 : FF
SUM: F2 AD 6F FC 86 C9 4B 80 36A3

```

```

3280 21 00 45 CD 89 36 11 00 : 03
3288 45 21 DF 33 CD 93 36 DD : FA
3290 E5 D1 21 00 45 CD AF 36 : CE
3298 E5 DD E1 3E 01 11 00 45 : 38
32A0 CD BC 40 21 00 45 DA FB : A4
32A8 35 2A 80 45 22 AE 45 3E : 07
32B0 02 32 53 45 CD 28 37 DD : D5
32B8 E5 E1 11 00 5F B7 ED 52 : 2C
32C0 38 0F 3A 3C 45 A7 20 89 : D2
32C8 21 AA 33 CD 17 37 C3 6D : 19
32D0 36 FD 7E 00 A7 21 CB 33 : 77
32D8 C4 77 36 20 2A 76 1F FD : C8
32E0 CD 17 37 FD 2A 76 1F FD : D4
32E8 36 00 00 3A 3C 45 A7 CA : 62
32F0 63 30 CD 40 3A ED 5B 42 : 6B
32F8 45 2A 56 45 B7 ED 52 21 : 21
SUM: 17 66 F5 E3 5B 2E 27 C6 F212

```

```

3300 E3 33 ED 4B 42 45 ED 5B : 1D

```

```

3308 56 45 1B C4 5D 33 ED 5B : 52
3310 44 45 2A 58 45 B7 ED 52 : 46
3318 21 EF 33 ED 4B 44 45 ED : F1
3320 5B 58 45 1B C4 5D 33 ED : 54
3328 5B 46 45 2A 5A 45 B7 ED : 53
3330 52 21 FB 33 ED 4B 46 45 : 64
3338 ED 5B 5A 45 1B C4 5D 33 : 56
3340 3A 28 45 A7 28 14 5E 04 : CC
3348 11 28 45 CD A6 11 21 28 : 78
3350 45 DA FB 3C CD 28 30 CD : 4E
3358 FC 41 C3 FA 1F CD 17 37 : 34
3360 60 69 CD BE 1F 21 07 34 : CF
3368 CD 17 37 62 6B CD BE 1F : 92
3370 CD EE 1F C9 2F 50 3A 00 : 5C
3378 2F 70 3A 00 2F 44 3A 00 : 86
SUM: 48 0F E9 9D F7 F0 85 CA 0045

```

```

3380 2F 64 3A 00 2F 53 00 2F : 7E
3388 73 00 2F 40 80 2F 60 00 : B5
3390 2F 4E 3A 50 00 2F 6E 3A : DE
3398 70 00 2F 4E 3A 44 00 2F : 3A
33A0 6E 3A 44 00 2F 4E 00 2F : B9
33A8 6E 00 53 6F 72 79 20 : AD
33B0 21 20 63 6F 6D 6D 61 6E : BC
33B8 64 73 20 61 72 65 20 74 : C3
33C0 6F 6F 20 6D 61 6E 79 20 : D3
33C8 21 00 00 2C 00 3F 45 72 : 50
33D0 72 00 00 4C 49 42 00 4D : A3
33D8 41 50 00 4F 42 4A 00 52 : BE
33E0 45 4C 00 43 53 45 47 20 : D3
33E8 61 72 65 61 20 3A 00 44 : 37
33F0 53 45 47 20 61 72 65 61 : 98
33F8 20 3A 00 57 53 45 47 20 : B0
SUM: FE 95 D8 79 FC F6 86 DF 0AB1

```

```

3400 61 72 65 61 20 3A 00 20 : 13
3408 2D 20 00 3E 2A CD F4 1F : 95
3410 ED 5B 76 1F CD C3 1F 1A : B6
3418 FE 1B CA 6D 36 13 1A A7 : 5A
3420 28 E9 5D FE 1F C9 DD E5 : 4F
3428 E1 CD 22 37 3F FF FE FF : 39
3430 C8 FE 0E C2 34 36 36 0E : E8
3438 23 CD 22 37 73 CD 22 D2 : D2
3440 37 77 23 CD 22 37 73 91 : 91
3448 A7 20 F8 18 CD 21 BC 35 : C5
3450 CD 17 37 3E 01 32 3D 45 : 0E
3458 CD 45 40 2A 42 45 22 56 : 7B
3460 45 2A 44 45 22 58 45 22 : D9
3468 5A 45 3E 01 11 14 45 CD : 15
3470 A6 41 21 14 45 DA FB 35 : 6B
3478 DD 21 00 50 DD 7E 00 DD : 86
SUM: 07 4D D3 4F A5 A1 22 DA EFBE

```

```

3480 23 FE FF CA 52 35 FE 01 : 70
3488 20 21 DD 6E 00 2E DD 69 : 69
3490 66 00 DD 23 ED 5B 56 45 : 49
3498 22 56 45 B7 ED 52 DA 7C : 09
34A0 3A 3A 60 45 A7 CA EC 35 : 9F
34A8 C3 7C 34 FE 02 20 24 DD : 94
34B0 6E 00 DD 23 DD 66 00 DD : 8E
34B8 23 ED 5B 58 45 22 5A 45 : C9
34C0 22 58 45 B7 ED 52 DA 7C : 0B
34C8 34 3A 61 45 A7 CA EC 35 : A0
34D0 C3 7C 34 FE 03 20 2E 21 : E3
34D8 C4 35 CD 17 37 DD 5E E1 : B7
34E0 CD 17 37 CD EE 1F 3E 01 : 34
34E8 DD E5 D1 CD BC 40 DD E5 : 1E
34F0 E1 DA FB 35 2A 80 45 22 : 2C
34F8 AE 45 3E 02 32 53 45 CD : CA
SUM: 69 76 B2 B2 CB A0 39 5B 23D0

```

```

3500 28 37 C3 7C 34 FE 04 C2 : 96
3508 7C 34 21 CA 35 CD 17 37 : E5
3510 DD E5 E1 CD 17 CD EE : 79
3518 1F 3E 01 DD E5 D1 CD BC : 7A
3520 40 DD E5 E1 DA FB 35 DD : CA
3528 7E 00 DD 23 A7 20 F8 DD : 1A
3530 6E 00 DD 23 DD 66 00 DD : SE
3538 23 7C B5 CA 7C 34 CD 82 : 1D
3540 42 2A 80 45 22 AE 45 3E : B4
3548 02 32 53 45 CD 28 37 C3 : B8
3550 2F 35 3A 61 45 A7 CA 9D : 52
3558 35 2A 5E 45 7C B5 CA 9D : 9A
3560 35 ED 5B 56 45 2A 44 45 : CE
3568 37 ED 52 DA 7D 35 2A 44 : 70
3570 45 ED 5B 56 45 B7 ED 52 : 1E
3578 CD EC 35 18 0B 2A 5E 45 : DE
SUM: 15 55 F2 A9 01 FA 78 17 0A9D

```

```

3580 7C B5 21 CD 35 C4 17 37 : 8F
3588 ED 4B 5E 45 21 00 90 7E : 8A
3590 23 C5 E5 CD C1 42 E1 C1 : 3F
3598 0B 78 B1 20 F2 2A 42 44 : 5F
35A0 3A 60 45 A7 20 03 2A 44 : 1F
35A8 45 22 00 46 3A 62 45 A7 : 35
35B0 28 03 2A 63 45 22 02 46 : 87
35B8 CD FC 41 C9 50 41 50 53 : 8A
35C0 2D 32 0D 00 4C 65 65 68 : 8A
35C8 69 6E 67 20 00 43 53 45 : 39
35D8 47 20 61 6E 64 20 14 53 : 50
35E0 69 65 65 64 20 75 70 20 : E3
35E8 21 21 00 00 7C B5 C8 E5 : 00
35F0 AF CD C1 42 E1 DA 19 3E : 86
35F8 2B 18 F1 E5 21 0B 3C CD : 46
SUM: 91 37 DE 92 B8 38 3A BA E07E

```

```

3600 17 37 E1 CD 17 37 CD EE : 0E
3608 1F 18 62 43 61 6E 20 6E : 39

```

```

3610 6F 74 20 6F 70 65 6E 20 : D5
3618 00 21 21 36 CD 17 37 18 : AB
3620 40 66 69 6C 65 20 41 63 : B0
3628 63 65 73 73 20 45 72 72 : F7
3630 6F 72 0D 00 21 3B 36 CD : 1D
3638 17 37 C9 49 6C 6C 65 67 : 04
3640 61 6C 20 4C 49 42 20 46 : 2A
3648 69 6C 65 20 45 72 72 6F : F2
3650 72 0D 00 21 5A 36 CD 17 : 14
3658 37 C9 49 6C 6C 65 67 61 : AE
3660 6C 20 4F 52 17 20 45 72 : 1R
3668 72 6F 72 0D 00 3A 3D 45 : 1C
3670 A7 C4 FC 41 C3 FA 1F FD : M1
3678 E5 D1 7E A7 28 07 1A 8E : F2
SUM: B7 2A 3F 1D 4D D7 61 3C B5CF

```

```

3680 C0 13 23 18 F5 D5 FD E1 : B5
3688 C9 7E 23 A7 37 C8 FE 2E : 3C
3690 20 F7 C9 06 0F 1A CD DE : BA
3698 36 38 09 05 28 06 FE 2E : 16
36A0 C0 13 18 F1 FE 25 C8 3E : 16
36A8 2E 12 13 CD AF 36 C9 06 : D4
36B0 14 AF 12 7E 23 CD DE 36 : 57
36B8 38 03 05 28 06 12 13 AF : 48
36C0 12 1R F0 E9 23 C9 06 11 : 0B
36C8 AF 12 7E CD DE 36 D8 05 : FD
36D0 C8 23 12 13 AF 12 18 F2 : DB
36D8 CD 17 37 C3 6D 36 FE 2F : AE
36E0 28 05 FE 2C 28 02 A7 C0 : E9
36E8 37 C9 21 00 00 FD 7E 00 : 9C
36F0 16 30 FE 30 D8 FE 3A 38 : BC
36FF 11 16 37 FE 41 D8 FE 47 : BA
SUM: FD 16 65 16 97 1C 99 BD 228E

```

```

3700 38 08 16 57 FE 61 D8 FE : E2
3708 61 00 19 29 29 29 92 16 : 83
3710 00 5F 19 FD 23 18 06 7E : 04
3718 A7 C8 23 E5 CD F4 1F E1 : 38
3720 1E F5 E5 CD 6E 42 E1 C9 : 19
3728 CD 5E 42 32 50 45 FE FF : 41
3730 C8 21 28 37 E5 DD 73 65 : F2
3738 45 E6 0E CA 9D 38 FE 20 : C8
3740 CA 06 38 FE 60 CA 10 39 : 3B
3748 3A 50 45 E6 F8 FE 90 CA : 05
3750 E5 3E FE 98 CA 00 3A FE : BF
3758 A0 CA 12 3A FE A8 CA 2B : 51
3760 3A 3A 50 45 E6 7F 26 00 : 94
3768 6F 20 11 73 37 19 7E 23 : 0D
3770 6F 5F 59 22 39 5F 39 89 : 3A
3778 6F 5F 39 45 39 AD 39 DA : A7
SUM: 12 ED BA 37 06 56 69 72 6A72

```

```

3780 38 E7 39 E5 3E E5 3E E5 : 84
3788 38 E5 3E E5 3E E5 3E E5 : 8C
3790 3E E5 3E 73 38 73 38 73 : 2A
3798 38 73 38 73 38 73 38 73 : AC
37A0 38 73 38 73 38 73 38 73 : AC
37A8 38 73 38 73 38 73 38 73 : AC
37B0 38 73 38 73 38 73 38 73 : AC
37B8 38 73 38 73 38 73 38 73 : AC
37C0 38 73 38 73 38 73 38 73 : AC
37C8 38 73 38 2E 3A E5 3E E5 : 53
37D0 38 E5 3E E5 3E E5 3E E5 : 8C
37D8 38 E5 3E E5 3E E5 3E E5 : 8C
37E0 38 E5 3E E5 3E E5 3E E5 : 8C
37E8 38 E5 3E 3D 3A 48 3A 55 : AF
37F0 3A 82 3A 6F 3A 79 3A 82 : B4
SUM: A7 3F A7 EB A4 B7 A8 CD BFF0

```

```

3800 3A 80 3A 9E 3A B8 3A 89 : 97
3808 3A 07 3A E4 3A F4 3A 04 : 9B
3810 3E 14 38 E5 3E E5 3E E5 : 8C
3818 3E E5 3E E5 3E E5 3E E5 : 8C
3820 3E E5 3E E5 3E E5 3E E5 : 8C
3828 3E E5 3E 73 38 1E 3B 32 : 97
3830 3E E5 3E 43 3B 4B 3B 52 : 1A
3838 3E 40 3B 67 38 81 3B E5 : 09
3840 3E E5 3E E5 3E E5 3E E5 : 8C
3848 3E E5 3E E5 3E E5 3E E5 : 8C
3850 3E E5 3E E5 3E E5 3E E5 : 8C
3858 3E E5 3E E5 3E E5 3E E5 : 8C
3860 3E AD 3B 97 3B 9E 3B A5 : 56
3868 20 73 38 11 85 38 CD E5 : 66
3870 1F 1A 50 45 CD C1 1F CD : 68
SUM: A7 7E D8 B4 9F 55 3C B5 AE0F

```

```

3880 E0 1F C3 FA 1F 49 6E 74 : 14
3888 72 E5 61 6C 20 45 72 E5 : A2
3890 72 A7 72 20 19 54 45 72 : BF
3898 20 45 6F 3A 00 CD 8F 3E : AF
38A0 E0 78 AE 45 19 E5 ED 5B : 81
38A8 50 45 5F ED 52 F1 38 03 : B1
38B0 2E 50 45 E5 3A 50 45 E6 : 81
38B8 1F 47 7D 50 3C E1 3A 3D : 24
38C0 47 3E 01 CD AD 3B C9 : A8
38C8 3E 45 E6 1F 47 CD 5D : 45
38D0 3C 7D 14 3C 30 18 F5 CD : 63
38D8 3A 3E 21 3E 00 CC AD 3B : B4
38E0 3E 01 32 52 45 3A 3D 45 : C4
38E8 A7 04 2A 3F 18 1E E5 11 : 00
38F0 80 50 12 7E A7 E1 20 0C : AB
38F8 3E 01 32 52 45 3A 3D 45 : C4
SUM: 4E 0D B5 EB 19 CC 54 C7 1FF0

```

```

3900 A7 CD F5 3E 29 11 00 70 : 46
3908 1E 88 23 56 CD F1 3D C9 : B4
3910 3A 50 45 E6 1F 3C 47 C5 : 1C

```


3918 CD 6E 42 CD 37 3E C1 10 : 90
3920 F6 C9 CD 8F 3E ED 5B AE : 4F
3928 45 19 E5 ED 5B B0 45 B7 : 37
3930 ED 52 E1 38 03 72 80 45 : 72
3938 E5 11 00 60 19 7E F1 A7 : 75
3940 20 10 3E 01 32 52 45 3A : 72
3948 3D 45 A7 28 05 CD F5 3E : 56
3950 18 09 11 00 70 29 19 7E : 62
3958 23 66 6F CD E9 3D C9 CD : 81
3960 8F 3E ED 5B AE 45 19 E5 : 06
3968 ED 5B B0 45 B7 ED 52 E1 : 14
3970 38 03 22 B0 45 E5 11 00 : 48
3978 60 19 36 01 E1 29 11 00 : CB
SUM: 80 9C 8C A2 1C 7E 1F E8 AC0D

3980 70 19 CD 0D 3E 73 23 72 : A9
3988 C9 CD 19 3E E5 7D CD 37 : 53
3990 3E E1 7C CD 37 3E C9 CD : 73
3998 19 3E E5 7C CD 37 3E E1 : DB
39A0 7D CD 37 3E C9 CD 19 3E : AC
39A8 7D CD 37 3E C9 CD 9A 3E : 2D
39B0 44 4D CD 19 3E 3A 3D 45 : 71
39B8 A7 CA 37 3E 37 ED B7 F2 : 84
39C0 0B 24 25 20 08 7D B7 F2 : A9
39C8 37 3E 18 08 7D 24 20 05 : 5B
39D0 B7 FA 37 3E CD 37 3E C9 : 3E
39D8 19 3F CD 9A 3E E5 CD 8F : 3E
39E0 3E D1 19 CD ED 3D C9 CD : A5
39E8 9A 3E CD E9 3D C9 3A 50 : 1E
39F0 45 E6 07 3C 47 C5 CD 6E : B5
39F8 42 CD 37 3E C1 10 F6 C9 : 14
SUM: E6 13 1E 97 8F BA B6 65 E416

3A00 3A 50 45 E6 07 3C 47 C5 : 04
3A08 CD 8F 3E CD DE 3D C1 10 : 53
3A10 F6 C9 3A 50 45 E6 07 3C : B7
3A18 47 C5 CD 8F 3E E5 7C CD : D4
3A20 37 3E E1 7D CD 37 3E C1 : D6
3A28 10 EF C9 C3 E2 39 CD 19 : 98
3A30 3E 7C B5 C8 E5 AF CD 37 : CF
3A38 3E E1 2B 18 F4 CD 11 3E : 72
3A40 CD 1D 3E 19 CD E9 3D C9 : FD
3A48 CD 11 3E CD 1D 3E B7 ED : E8
3A50 52 CD E9 3D C9 CD 11 3E : 2A
3A58 CD 1D 3E CD E9 3D C9 CD : 54
3A60 3D C9 CD 11 3E CD 1D 3E : 4A
3A68 CD 7C 40 CD E9 3D C9 CD : 12
3A70 1D 3E 6C 26 00 CD E9 3D : E0
3A78 C9 CD 1D 3E 26 00 CD E9 : CD
SUM: B0 5F 4D E4 65 3B E2 3B 779C

3A80 3D C9 CD 1D 3E 7C 2F 67 : 40
3A88 7D 2F 6F 23 CD E9 3D C9 : FA
3A90 CD 11 3E CD 1D 3E CD 7C : 8D
3A98 40 EB CD E9 3D C9 CD 1D : D1
3AA0 3E CD 11 3E 44 4D 21 81 : 0D
3AA8 00 78 B1 CA E9 3D C5 D5 : B3
3AB0 CD 69 40 CD 11 08 18 F1 : 1C
3AB8 CD 11 3E CD 1D 3E C8 3C : 4B
3AC0 CB 1D 1D 20 F9 CD E9 3D : 11
3AC8 C9 CD 11 3E CD 1D 3E 29 : 36
3AD0 1D 20 FC CD E9 3D C9 CD : C2
3AD8 1D 3E 7C 2F 67 7D 2F 6F : 88
3AE0 CD E9 3D C9 CD 11 3E CD : A5
3AE8 1D 3E 7C 2F 67 7D B3 6F : 8F
3AF0 CD E9 3D C9 CD 11 3E CD : A5
3AF8 1D 3E 7C B2 67 7D B3 6F : 8F
SUM: 41 49 9F EC EE FF D0 E6 14A7

3B00 CD E9 3D C9 CD 11 3E CD : A5
3B08 1D 3E 7C AA 67 7D AB 6F : 7F
3B10 CD E9 3D C9 CD 1D 3E 7C : C0
3B18 65 6F CD E9 3D C9 CD 1D : 6B
3B20 42 06 02 47 28 07 06 03 : 29
3B28 3D 28 02 06 04 78 32 53 : 8E
3B30 45 C9 CD 19 3E 22 54 45 : ED
3B38 3E 01 32 51 45 C9 AF 32 : B1
3B40 51 45 C9 CD 19 3E 7D CD : CD
3B48 37 3E C9 CD 19 3E CD DE : 0D
3B50 3D C9 CD 19 3E E5 7C CD : 58
3B58 37 3E E1 7D CD 37 3E C9 : DE
3B60 CD 8F 3E CD E9 3D C9 CD : 23
3B68 9A 3E EB CD 19 3E B7 ED : 8B
3B70 52 C8 DA 3F C3 E5 AF CD : D3
3B78 37 3E E1 2B 7C B5 20 F5 : C7
SUM: 0A D4 EA 70 E7 8B 82 B0 05CD

3B80 C9 CD 1D 3E 22 63 45 3E : F9
3B88 01 32 62 45 C9 CD 6E 42 : 20
3B90 A7 C8 CD F4 1F 18 F6 CD : 2A
3B98 1D 3E CD E9 3F C9 CD 1D : 03
3BA0 3E CD BE 1F C9 CD 1D 3E : D9
3BA8 7D CD C1 1F C9 22 11 3C : 62
3BB0 32 13 3C CD 14 3C D2 1F : 8F
3BB8 3F 20 42 CD 6E 3C 26 00 : 3E
3BC0 6F 29 44 4D 6A 6F CD AF : C3
3BC8 40 78 B1 20 7F 2B 2B ED : CE
3BD0 4B 8B 45 CD 9D 40 2A 8B : 7A
3BD8 45 01 00 00 CD 9D 40 3A : 2A
3BE0 13 3C CD 98 40 ED 4B 11 : 3D
3BE8 3C CD 9D 40 11 8D 45 1A : E3
3BF0 13 F5 CD 98 40 F1 A7 20 : 65
3BF8 F6 22 8B 45 C9 2A 5B 3C : 72
SUM: 51 1F 12 27 78 7E 90 EB 17BB

3C00 23 23 3A 13 3C CD 98 40 : 74
3C08 ED 4B 11 3C CD 9D 40 B7 : E6
3C10 C9 00 00 00 CD 6E 3C 26 : 66
3C18 00 6F 29 22 5B 3C 2A 5B : D6
3C20 3C CD AF 40 78 B1 37 C8 : 20
3C28 ED 43 5B 3C 21 05 00 09 : F6
3C30 11 8D 45 1A A7 28 0A 13 : E9
3C38 47 CD AA 40 B8 28 F4 18 : EA

3C40 DD CD 94 1F A7 20 D7 2A : 25
3C48 5B 3C 23 23 CD AA 40 CD : 61
3C50 AF 40 60 69 B7 C0 3E 01 : 6E
3C58 B7 37 C9 00 00 21 8D 45 : AA
3C60 C5 B5 CD 6E 42 E1 C1 77 : 40
3C68 23 10 F5 36 00 C9 F5 C5 : D1
3C70 21 8D 45 06 00 7E 23 A7 : 41
3C78 28 04 80 47 18 F7 78 C1 : 3B
SUM: 29 4D D4 E3 AE E4 96 55 1054

3C80 E1 C9 21 00 00 01 01 02 : CF
3C88 AF CD 9A 1F ED A1 EA 89 : 36
3C90 3C C9 AF 32 67 45 21 00 : B3
3C98 00 22 68 45 21 00 02 ED : DF
3CA0 5B 8B 45 E5 B7 ED 52 E1 : E7
3CA8 30 14 23 23 CD AA 40 A7 : E8
3CB0 20 05 CD E6 3C 18 E8 23 : 37
3CB8 23 CD 21 3D 18 E1 CD EB : FF
3CC0 1F 2A 68 45 7C B5 C8 CD : BC
3CC8 E9 3F 21 D1 3C CD 17 37 : 71
3CD0 C9 20 55 6E 64 65 66 69 : 44
3CD8 6E 65 64 20 4C 61 62 65 : CB
3CE0 6C 28 73 29 0D 00 3E 2D : A8
3CE8 CD F4 1F CD AF 40 C5 06 : 67
3CF0 0A CD AA 40 A7 28 08 CD : 65
3CF8 F4 1F 10 F5 CD 21 3D 04 : 47
SUM: 10 E8 B6 90 E5 48 44 E4 BAB2

3D00 CD F1 1F 10 FB E3 CD BE : 56
3D08 1F CD F1 1F 2A 68 45 23 : F6
3D10 22 68 45 3A 67 45 3C E5 : D7
3D18 03 32 67 45 C0 EB 1F E1 : 98
3D20 C9 CD AA 40 A7 C8 18 F9 : 00
3D28 AF 32 67 45 21 00 02 E5 : 95
3D30 ED 5B 8B 45 B7 ED 52 E1 : EF
3D38 D2 73 3D 23 23 3E 2D CD : 00
3D40 D4 3D CD AA 40 A7 28 1A : B1
3D48 CD AF 40 C5 CD 82 3D E3 : F0
3D50 29 11 00 70 19 7E 23 66 : CA
3D58 6F CD B7 3D CD 9B 3D E1 : B6
3D60 18 CD 23 23 CD 82 3D 11 : C9
3D68 7D 3D CD AE 3D CD 9B 3D : 17
3D70 3C 2F 3D 3E 0D CD 2A 3D : 58
3D78 AF CD D4 3D C9 2A 24 D4 : D4
SUM: 88 F5 5A 03 CD F6 A1 2D A46D

3D80 2A 00 06 0A CD AA 40 A7 : 98
3D88 28 08 CD D4 3D 10 F5 CD : E0
3D90 21 3D 04 3E 20 CD D4 3D : 9E
3D98 10 F9 C9 3A 67 45 3C E6 : DA
3DA0 03 32 67 45 3E 20 C2 D4 : D5
3DA8 3D 3E 0D C3 D4 3D 1A A7 : 1D
3DB0 CD 13 CD CD D4 3E 1F 7C : 44
3DB8 CD B7 3D 7D F5 0F 0F 0F : 65
3DC0 0F CD E5 3D F1 CD 0A 3D : A3
3DC8 18 0A E6 0F F6 30 F5 3A : 75
3DD0 08 C6 07 C9 C5 D5 E5 CD : BA
3DD8 C1 42 E1 D1 C1 C9 E5 7D : A1
3DE0 CD 37 3E E1 7C CD 37 3E : E1
3DE8 C9 D5 54 5D CD F1 3D 3D : 1B
3DF0 C9 3A 6A 45 FE 11 D2 35 : C8
3DF8 3F E5 D5 26 00 FE 29 11 : C8
SUM: B6 87 82 3E 89 29 28 B3 E215

3E00 6B 45 19 D1 73 23 72 E1 : 83
3E08 3C 32 6A 45 C9 AF 32 52 : 19
3E10 45 E5 CD 1D 3E 54 5D E1 : E4
3E18 C9 AF 32 52 45 3A 6A 45 : 2A
3E20 A7 CA 3A 3F 3D 32 6A 45 : 08
3E28 D5 26 00 6F 29 11 68 45 : 54
3E30 19 7E 23 66 6F D1 C9 CD : F6
3E38 B5 3E 26 00 6F 3A 3D 45 : 44
3E40 A7 C8 3A 53 45 FE 03 28 : 6A
3E48 0C FE 04 C8 3A 60 45 A7 : 5C
3E50 7D C4 C1 42 C9 3A 61 45 : ED
3E58 A7 C8 E5 2A 5E 45 23 22 : 66
3E60 5E 45 7C 11 00 90 19 D1 : AA
3E68 FE 20 30 02 73 C9 21 77 : 24
3E70 3E CD 17 37 C9 6D 36 44 : 03
3E78 53 45 47 20 62 75 66 66 : A2
SUM: C3 80 F3 8A 41 C6 E8 1D FFA3

3E80 65 72 20 69 73 66 76 76 : D8
3E88 65 72 20 21 21 00 00 CD : 13
3E90 6E 42 F5 CD 6E 42 67 F1 : 7A
3E98 6F C9 2A 54 45 3A 51 45 : CB
3EA0 A7 C0 2A 56 45 3A 53 45 : FE
3EA8 FE 02 C8 2A 58 45 FE 03 : 90
3EB0 C8 2A 5A 45 C9 F5 E5 3A : 6E
3EB8 53 45 FE 03 28 0D FE 04 : D0
3EC0 28 12 2A 56 45 23 22 56 : 9A
3EC8 45 18 10 2A 58 45 23 22 : 79
3ED0 58 45 18 07 2A 5A 45 23 : A8
3ED8 22 5A 45 2A 54 45 23 22 : C9
3EE0 54 45 E1 F1 C9 11 4B 3F : CF
3EE8 CD E5 1F 3A 50 45 CD C1 : 2E
3EF0 1F CD EE 1F C9 E5 11 5A : 12
3EF8 3F CD E5 1F E1 CD BE 1F : 9B
SUM: CD AD 13 8D B3 39 EF 35 C56E

3F00 CD EE 1F C9 E5 11 6D 3F : 45
3F08 CD E5 1F C9 E5 1F CD BE : 29
3F10 EE 1F C9 11 84 3F C3 44 : B1
3F18 3F 11 94 3F C3 11 84 3F : 7A
3F20 9C 3F CD E5 1F 11 8D 45 : 8F
3F28 18 1A 11 B1 3F CD E5 1F : 04
3F30 11 8D 45 18 0F 11 2C 3F : 1C
3F38 18 0A 11 D1 3F 18 05 11 : 71
3F40 DD 3F 18 00 CD E5 1F CD : D2
3F48 EE 1F C9 55 6E 64 65 66 : C8
3F50 6E 65 64 20 49 54 45 : A2
3F58 4D 00 55 6E 64 65 66 69 : A8
3F60 6E 65 64 20 4C 61 62 65 : CB

3F68 6C 2D 4E 6F 00 4D 75 6C : 84
3F70 74 69 20 44 65 66 69 6E : E3
3F78 65 64 20 4C 61 62 65 6C : C9
SUM: D8 1E 5C BF 76 C6 AA A1 29F6

3F80 2D 4E 6F 00 54 6F 6F 20 : 3C
3F88 4D 61 6E 79 20 4C 61 62 : C1
3F90 65 6C 73 00 54 6F 6F 20 : 96
3F98 46 61 72 00 4D 75 6C 74 : BB
3FA0 69 20 44 65 66 69 6E 65 : D4
3FA8 64 20 4C 61 62 65 6C 2D : 91
3FB0 00 55 6E 64 65 66 69 6E : 36
3FB8 65 64 20 4C 61 62 65 6C : C9
3FC0 2D 00 53 74 61 63 6B 20 : 43
3FC8 4F 76 65 72 66 6C 6F 77 : 54
3FD0 00 53 74 61 63 6B 20 45 : 5B
3FD8 6D 70 74 79 00 49 6C 6C : EB
3FE0 65 67 61 6C 20 4F 52 47 : A1
3FE8 00 D5 F5 AF 32 4F 45 06 : 45
3FF0 05 11 4F 45 0E 0A CD 13 : A2
3FF8 40 C6 30 1B 12 10 F5 6B : D3
SUM: EA C1 55 2A 3F 70 12 95 F539

4000 62 06 04 7E FE 30 20 05 : 3D
4008 36 20 23 18 F6 CD E5 1F : 58
4010 F1 E1 C9 C5 AF 06 10 29 : 4E
4018 17 2C 91 30 02 2D 81 10 : C4
4020 F6 C1 C9 AF 32 51 45 32 : 29
4028 52 45 32 62 45 21 00 02 : 93
4030 22 8B 45 01 00 23 21 00 : 24
4038 60 36 00 23 0B 78 B1 20 : 0D
4040 F8 CD 82 3C C9 AF 32 6A : 97
4048 45 32 8D 32 3D 32 95 42 : CD
4050 21 00 00 22 3D 32 95 42 : 9C
4058 45 22 56 45 22 58 45 22 : E8
4060 5A 45 22 54 45 22 55 45 : F1
4068 C9 44 2D 21 00 00 3E 10 : C9
4070 29 CB 23 CB 12 30 01 09 : 2E
4078 3D 20 F5 C9 42 4B 54 5D : 59
SUM: 96 8F FD AE 98 45 CC E8 D220

4080 3E 10 26 00 CB 23 CB 12 : 3F
4088 29 E5 B7 ED 42 E1 CB 03 : 10
4090 ED 42 13 3D 20 EE EB C9 : 41
4098 CD 9A 1F 23 C9 F5 79 CD : AD
40A0 9A 1F 23 78 CD 9A 1F 23 : FD
40A8 F1 C9 CD 94 1F 23 C9 F5 : 1B
40B0 CD 94 1F 4F 23 CD 94 1F : 72
40B8 47 23 F1 C9 CD A3 1F 08 : 8B
40C0 CD 25 41 D8 2A 4F 1F 11 : D9
40C8 28 46 01 20 00 ED B0 3A : 61
40D0 5D 1F 32 53 46 AF 32 84 : DC
40D8 45 CD A9 43 D8 06 10 0E : FA
40E0 00 3A 41 46 11 43 46 12 : 8D
40E8 13 FE 7F 30 0F 2A 62 1F : 7A
40F0 65 6F 30 01 24 7E 05 28 : F4
40F8 28 0C 18 EB 0D F5 78 87 : 39
SUM: 12 7A 3A 61 6B 0A 39 A7 3C0E

4100 87 87 87 87 AF F1 D6 80 81 : AC
4108 32 59 46 AF 32 54 46 01 : 4D
4110 37 00 11 B5 45 21 23 46 : CC
4118 ED B0 3E FF 32 95 42 B7 : 9A
4120 C9 3E 07 37 C9 3A 5D 1F : C4
4128 CD 44 41 D8 CD 4F 41 D8 : 5F
4130 3E 08 37 C0 E5 ED 5B 74 : DE
4138 1F 01 20 00 ED B0 E1 7E : 3C
4140 CD 99 41 C9 FE 41 38 04 : EA
4148 FE 45 3F D0 3E 03 C9 0E : 6B
4150 10 ED 5B 60 1F ED 53 55 : 6C
4158 46 2A 64 1F 3E 01 CD 00 : FF
4160 20 D8 08 08 22 57 46 7E : 4E
4168 FE FF 28 1A B7 28 0B D5 : FE
4170 ED 5B 74 1F CD 8A 41 D1 : 44
4178 28 0D D5 11 20 00 19 D1 : 25
SUM: 24 4F 71 EB 61 41 D1 C4 5EAB

4180 18 E2 13 0D 20 CF 3E AF : EE
4188 B7 C9 C5 E5 06 10 13 23 : 76
4190 1A BE 20 02 10 FE E1 C1 : A4
4198 C9 E5 E6 87 21 1F 28 E5 : 42
41A0 E1 C8 3E 06 37 C9 CD A3 : 5D
41A8 1F D8 CD AF 1F D8 01 20 : 8B
41B0 00 11 23 46 2A 74 1F ED : 24
41B8 0A 5A 5D 1F 32 53 46 2A : 5B
41C0 E1 27 22 57 46 2A DF 27 : F7
41C8 22 55 46 CD D8 43 D8 32 : AF
41D0 41 46 32 43 46 3E 80 32 : 32
41D8 44 46 AF 32 59 46 32 84 : F0
41E0 45 32 54 46 21 00 00 22 : 54
41E8 35 46 01 37 00 11 EC 45 : F5
41F0 21 23 46 ED B0 3E 80 32 : 97
41F8 DD 42 B7 C9 21 EC 15 11 : 02
SUM: 5A 1F 04 61 B8 8A 28 14 B113

4200 23 46 01 37 00 ED B0 3A : 78
4208 53 46 32 5D 1F 2A 35 46 : EC
4210 2C 2D C4 F0 42 3A 59 46 : 28
4218 2A 35 46 2C 2D 01 3C 5B : 5B
4220 67 22 35 46 3E 01 ED 5B : 8B
4228 55 46 2A 64 1F CD 00 20 : 35
4230 D8 21 23 46 ED 5B 57 46 : 47
4238 01 20 00 ED B0 3E 01 ED : EA
4240 5B 55 46 2A 64 1F CD 03 : 7A
4248 20 D8 CD A9 43 D8 06 10 : 9F
4250 21 43 46 7E FE 7F 30 12 : E7
4258 23 4E E5 2A 62 1F 00 17 : 04
4260 5F 19 71 E1 05 CA 21 41 : FB
4268 18 E9 CD C7 43 C9 21 95 : 57
4270 42 3A 20 08 CD 97 42 D8 : 1C
4278 21 E6 45 3A 2A 95 42 7E : FF
SUM: FA 71 A0 EC CE 2C 63 01 A6F9

▶この前、電器屋さんでMZ-2200の中古がデータレコーダとセットで8,000円で、売られていました。別のところでは初代PC-8001が本体のみ2,500円で売られてました。時の流れは速い。
本間 晃(19)愛知県

4500 10 20 30 40 50 60 70 80 90 100

2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032 2033 2034 2035 2036 2037 2038 2039 2040 2041 2042 2043 2044 2045 2046 2047 2048 2049 2050 2051 2052 2053 2054 2055 2056 2057 2058 2059 2060 2061 2062 2063 2064 2065 2066 2067 2068 2069 2070 2071 2072 2073 2074 2075 2076 2077 2078 2079 2080 2081 2082 2083 2084 2085 2086 2087 2088 2089 2090 2091 2092 2093 2094 2095 2096 2097 2098 2099 2100 2101 2102 2103 2104 2105 2106 2107 2108 2109 2110 2111 2112 2113 2114 2115 2116 2117 2118 2119 2120 2121 2122 2123 2124 2125 2126 2127 2128 2129 2130 2131 2132 2133 2134 2135 2136 2137 2138 2139 2140 2141 2142 2143 2144 2145 2146 2147 2148 2149 2150 2151 2152 2153 2154 2155 2156 2157 2158 2159 2160 2161 2162 2163 2164 2165 2166 2167 2168 2169 2170 2171 2172 2173 2174 2175 2176 2177 2178 2179 2180 2181 2182 2183 2184 2185 2186 2187 2188 2189 2190 2191 2192 2193 2194 2195 2196 2197 2198 2199 2200 2201 2202 2203 2204 2205 2206 2207 2208 2209 2210 2211 2212 2213 2214 2215 2216 2217 2218 2219 2220 2221 2222 2223 2224 2225 2226 2227 2228 2229 2230 2231 2232 2233 2234 2235 2236 2237 2238 2239 2240 2241 2242 2243 2244 2245 2246 2247 2248 2249 2250 2251 2252 2253 2254 2255 2256 2257 2258 2259 2260 2261 2262 2263 2264 2265 2266 2267 2268 2269 2270 2271 2272 2273 2274 2275 2276 2277 2278 2279 2280 2281 2282 2283 2284 2285 2286 2287 2288 2289 2290 2291 2292 2293 2294 2295 2296 2297 2298 2299 2300 2301 2302 2303 2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 2319 2320 2321 2322 2323 2324 2325 2326 2327 2328 2329 2330 2331 2332 2333 2334 2335 2336 2337 2338 2339 2340 2341 2342 2343 2344 2345 2346 2347 2348 2349 2350 2351 2352 2353 2354 2355 2356 2357 2358 2359 2360 2361 2362 2363 2364 2365 2366 2367 2368 2369 2370 2371 2372 2373 2374 2375 2376 2377 2378 2379 2380 2381 2382 2383 2384 2385 2386 2387 2388 2389 2390 2391 2392 2393 2394 2395 2396 2397 2398 2399 2400 2401 2402 2403 2404 2405 2406 2407 2408 2409 2410 2411 2412 2413 2414 2415 2416 2417 2418 2419 2420 2421 2422 2423 2424 2425 2426 2427 2428 2429 2430 2431 2432 2433 2434 2435 2436 2437 2438 2439 2440 2441 2442 2443 2444 2445 2446 2447 2448 2449 2450 2451 2452 2453 2454 2455 2456 2457 2458 2459 2460 2461 2462 2463 2464 2465 2466 2467 2468 2469 2470 2471 2472 2473 2474 2475 2476 2477 2478 2479 2480 2481 2482 2483 2484 2485 2486 2487 2488 2489 2490 2491 2492 2493 2494 2495 2496 2497 2498 2499 2500 2501 2502 2503 2504 2505 2506 2507 2508 2509 2510 2511 2512 2513 2514 2515 2516 2517 2518 2519 2520 2521 2522 2523 2524 2525 2526 2527 2528 2529 2530 2531 2532 2533 2534 2535 2536 2537 2538 2539 2540 2541 2542 2543 2544 2545 2546 2547 2548 2549 2550 2551 2552 2553 2554 2555 2556 2557 2558 2559 2560 2561 2562 2563 2564 2565 2566 2567 2568 2569 2570 2571 2572 2573 2574 2575 2576 2577 2578 2579 2580 2581 2582 2583 2584 2585 2586 2587 2588 2589 2590 2591 2592 2593 2594 2595 2596 2597 2598 2599 2600 2601 2602 2603 2604 2605 2606 2607 2608 2609 2610 2611 2612 2613 2614 2615 2616 2617 2618 2619 2620 2621 2622 2623 2624 2625 2626 2627 2628 2629 2630 2631 2632 2633 2634 2635 2636 2637 2638 2639 2640 2641 2642 2643 2644 2645 2646 2647 2648 2649 2650 2651 2652 2653 2654 2655 2656 2657 2658 2659 2660 2661 2662 2663 2664 2665 2666 2667 2668 2669 2670 2671 2672 2673 2674 2675 2676 2677 2678 2679 2680 2681 2682 2683 2684 2685 2686 2687 2688 2689 2690 2691 2692 2693 2694 2695 2696 2697 2698 2699 2700 2701 2702 2703 2704 2705 2706 2707 2708 2709 2710 2711 2712 2713 2714 2715 2716 2717 2718 2719 2720 2721 2722 2723 2724 2725 2726 2727 2728 2729 2730 2731 2732 2733 2734 2735 2736 2737 2738 2739 2740 2741 2742 2743 2744 2745 2746 2747 2748 2749 2750 2751 2752 2753 2754 2755 2756 2757 2758 2759 2760 2761 2762 2763 2764 2765 2766 2767 2768 2769 2770 2771 2772 2773 2774 2775 2776 2777 2778 2779 2780 2781 2782 2783 2784 2785 2786 2787 2788 2789 2790 2791 2792 2793 2794 2795 2796 2797 2798 2799 2800 2801 2802 2803 2804 2805 2806 2807 2808 2809 2810 2811 2812 2813 2814 2815 2816 2817 2818

リスト2

```

1 0000* : ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
2 0000* :
3 0000* : Wk Link Program Ver 1.00
4 0000* : Programed By T.Schigami
5 0000* : '90 Mar 7th
6 0000* :
7 0000* : CSEG 4000H -
8 0000* : DSEG 4500H -
9 0000* :
10 0000* :
11 0000* : ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
12 0000* :
13 0000* : CSEG
14 0000* :
15 0000* : NAME EQU 15
16 0000* : EXT EQU 3
17 0000* : LENGTH EQU NAME + 1 + EXT + 1
18 0000* :
19 0000* : CR EQU 00H
20 0000* : HEX EQU 10H
21 0000* : EOF EQU -1
22 0000* :
23 0001* : PRC EQU 1
24 0002* : DTA EQU 2
25 0003* : REL EQU 3
26 0004* : LIR EQU 4
27 0005* : MIB EQU 5
28 0006* :
29 0000* : C INCLUDE SCS.DEF
30 0001* : C BOT EQU 177A
31 0002* : C PRINT EQU 1FF4H
32 0003* : C PRINTS EQU 1FF7H
33 0004* : C JTNL EQU 1F72H
34 0005* : C JNL EQU 1F58H
35 0006* : C KSK EQU 1F5EH
36 0007* : C TAB EQU 1F70H
37 0008* : C GETL EQU 1F03H
38 0009* : C PAUSE EQU 1F77H
39 0010* : C JPNCHK EQU 1F71H
40 0011* : C PTRNL EQU 1F5EH
41 0012* : C HEX EQU 1F08H
42 0013* : C FILE EQU 1FA3H
43 0014* : C WOPEN EQU 1FAFH
44 0015* : C PORE EQU 1F9AH
45 0016* : C PEEX EQU 1F94H
46 0017* : C JOOPEN EQU 2000H
47 0018* : C KILL EQU 2015H
48 0019* : C NAME EQU 2012H
49 0020* : C ERROR EQU 2013H
50 0000* : C
51 0000* : C DREDSN EQU 2000H
52 0001* : C DWTSN EQU 2003H
53 0002* : C
54 0003* : C PRONT EQU 1F7AH
55 0004* : C KOPAD EQU 1F75H
56 0005* : C IOPAD EQU 1F74H
57 0006* : C SIZE EQU 1F72H
58 0007* : C MEMAI EQU 1F6AH
59 0008* : C JTPUP EQU 1F64H
60 0009* : C PATBF EQU 1F62H
61 0010* : C DIFPS EQU 1F60H
62 0011* : C FATPOS EQU 1F5F5H
63 0012* : C DSK EQU 1F50H
64 0000* : C
65 0000* : C INCLUDE WK.DEF
66 0000* : C
67 0000* : C Header File For Wk
68 0000* : C : CSEG 3000H-
69 0000* : C : DSEG 4500H-
70 0000* : C
71 0000* : C LBLMAX EQU 1000H
72 0001* : C
73 0002* : C Cmbuf EQU 5000H : - 5FFH
74 0003* : C Cmbloc EQU 5F00H
75 0004* : C
76 0005* : C LBLPLD EQU 6000H : - 6FFH
77 0006* : C LBLFNU EQU 7000H : - 8FFFH
78 0000* : C
79 0000* : C BF.DSEG EQU 3000H : - 4FFFH
80 0001* : C BDRBUF EQU 6000H : - 6FFFH
81 0002* : C WRBUF EQU 6010H : - BFFFH
82 0000* : C
83 0000* :
84 0000* : : External:
85 0000* :
86 0000* : : EXT LABOFF
87 0000* : : EXT KTIFFF
88 0000* : : EXT PLOCQ
89 0000* : : EXT FLOST
90 0000* : : EXT PLCEX
91 0000* : : EXT EXADR
92 0000* : : EXT PFLCD
93 0000* : : EXT PRDCT
94 0000* : : EXT PRAWK
95 0000* : : EXT CNTXT
96 0000* : : EXT Wknt
97 0000* : : EXT SEFMD
98 0000* :
99 0000* : : Start
100 0000* : main:
101 0000* : LD SP,(_REXAX)
102 0001* :
103 0002* : ED 5B 76 1F LD DE,(_REXAD)
104 0003* :
105 0004* : INC DE
106 0005* : INC DE : 'Skip 'a' or 'd'
107 0006* : LD B,1 : :B = src
108 0007* : CC1: LD A,(DE)
109 0008* : INC DE
110 0009* : AND A
111 0010* : JR JZ,C3
112 0011* : AND A
113 0012* : JR JP,NZ,CC1
114 0013* : INC B
115 0014* : JR CC1
116 0015* : IF (DE) = ' ' then arg
117 0016* : IF 64
118 0017* :
119 0018* :
120 0019* :
121 0020* :
122 0021* :
123 0022* :
124 0023* :
125 0024* :
126 0025* :
127 0026* :
128 0027* :
129 0028* :
130 0029* :
131 0030* :
132 0031* :
133 0032* :
134 0033* :
135 0034* :
136 0035* :
137 0036* :
138 0037* :
139 0038* :
140 0039* :
141 0040* :
142 0041* :
143 0042* :
144 0043* :
145 0044* :
146 0045* :
147 0046* :
148 0047* :
149 0048* :
150 0049* :
151 0050* :
152 0051* :
153 0052* :
154 0053* :
155 0054* :
156 0055* :
157 0056* :
158 0057* :
159 0058* :
160 0059* :
161 0060* :
162 0061* :
163 0062* :
164 0063* :
165 0064* :
166 0065* :
167 0066* :
168 0067* :
169 0068* :
170 0069* :
171 0070* :
172 0071* :
173 0072* :
174 0073* :
175 0074* :
176 0075* :
177 0076* :
178 0077* :
179 0078* :
180 0079* :
181 0080* :
182 0081* :
183 0082* :
184 0083* :
185 0084* :
186 0085* :
187 0086* :
188 0087* :
189 0088* :
190 0089* :
191 0090* :
192 0091* :
193 0092* :
194 0093* :
195 0094* :
196 0095* :
197 0096* :
198 0097* :
199 0098* :
200 0099* :
201 0100* :
202 0101* :
203 0102* :
204 0103* :
205 0104* :
206 0105* :
207 0106* :
208 0107* :
209 0108* :
210 0109* :
211 0110* :
212 0111* :
213 0112* :
214 0113* :
215 0114* :
216 0115* :
217 0116* :
218 0117* :
219 0118* :
220 0119* :
221 0120* :
222 0121* :
223 0122* :
224 0123* :
225 0124* :
226 0125* :
227 0126* :
228 0127* :
229 0128* :
230 0129* :
231 0130* :
232 0131* :
233 0132* :
234 0133* :
235 0134* :
236 0135* :
237 0136* :
238 0137* :
239 0138* :
240 0139* :
241 0140* :
242 0141* :
243 0142* :
244 0143* :
245 0144* :
246 0145* :
247 0146* :
248 0147* :
249 0148* :
250 0149* :
251 0150* :
252 0151* :
253 0152* :
254 0153* :
255 0154* :
256 0155* :
257 0156* :
258 0157* :
259 0158* :
260 0159* :
261 0160* :
262 0161* :
263 0162* :
264 0163* :
265 0164* :
266 0165* :
267 0166* :
268 0167* :
269 0168* :
270 0169* :
271 0170* :
272 0171* :
273 0172* :
274 0173* :
275 0174* :
276 0175* :
277 0176* :
278 0177* :
279 0178* :
280 0179* :
281 0180* :
282 0181* :
283 0182* :
284 0183* :
285 0184* :
286 0185* :
287 0186* :
288 0187* :
289 0188* :
290 0189* :
291 0190* :
292 0191* :
293 0192* :
294 0193* :
295 0194* :
296 0195* :
297 0196* :
298 0197* :
299 0198* :
300 0199* :
301 0200* :
302 0201* :
303 0202* :
304 0203* :
305 0204* :
306 0205* :
307 0206* :
308 0207* :
309 0208* :
310 0209* :
311 0210* :
312 0211* :
313 0212* :
314 021
```

```

120 0025' 22 *** LD (strcd),HL
122 0026' 22 *** LD (strcd),HL
123 0028' 22 *** LD (PTFRD),HL
124 0032' 22 *** LD (PTFRD),HL
125 0031' AF JDE A
126 0032' 32 *** LD (PnsA),A
127 0035' 32 *** LD (OmsnA),A
128 0036' 32 *** LD (mnmA),A
129 0037' 32 *** LD (snfd?),A
130 0038' 32 *** LD (sfcd),A
131 0041' 32 *** LD (sfgd),A
132 0044' D0 B1 25 58 LD IX,cdbuf ;? ? ? ? ? ? ? ? ? ?
;-----
133 0040' POP AF ;A = nrec
134 0045' F1 CP I
135 0046' 20 05 JR NZ,C0B
137 0040' C0 *** CALL GETL
138 0050' 15 11 JR CC7
139 0050'
140 0052' FD 2A 76 JF C0B: LD IV,(X)BUFAD
141 0056' FD 23 INC IV
142 0057' FD 23 INC IV ;Sk-p '0' or '1'
143 0058' FD 23 LD A,(IV)
144 0059' FD 20 C04: LD A,(IV)
145 0057' FD 20 CP ''
146 0061' 20 F7 JR NZ,C04
147 0063'
148 0063' FD 7E 00 CC7: LD A,(IV)
149 0065' AT AND A
150 0067' 20 06 JR NZ,C0B
151 0069' C0 *** CALL PTRMAP
152 0069' C0 *** CALL GETL
153 006F'
154 006F' C0B:
155 006F' PUSH IV LD ,Rennance
156 0072' FD 15 PUSH IV ;LD HL,X
157 0074' E1 POP HL
158 0075' C2 *** CALL Copy2
159 0070' IS PUSH HL ;LD IV,H
160 0078' FD E1 POP IV
161 0078'
162 0078' 21 *** LD HL,wP
163 007C' C2 *** CALL smatch
164 0081' 21 *** LD HL,wP+4
165 0084' C2 *** CALL smatch
166 0087' 20 34 JR NZ,C0B
167 0086'
168 0089' C0 *** CALL h1hex
169 008C' ES PUSH HL
170 0090' FD 58 *** LD DE,(PTFRD)
171 0091' F7 OR DE
172 0092' FD 52 SBC HL,DE
173 0094' E1 POP HL
174 0095' BC *** CALL C_EXR14 ;illegal REG Error
175 0095'
176 0095' 22 *** LD (PTFRD),HL
177 0095'
178 0095' D0 36 00 01 LD IX,PBG
179 009F' D0 23 INC IX
180 00A1' D0 75 04 LD (IX),I
181 00A4' D0 23 INC IX
182 00A6' D0 74 00 LD (IX),H
183 00A8' D0 23 INC IX
184 00A9'
185 00AB' 3A *** LD A,(sfgcd)
186 00AC' AT AND A
187 00AF' C2 *** JP NZ,C012
188 00B2'
189 00B2' 30 01 LD A,I
190 00B4' 32 *** LD (sfcd),A
191 00B7' 22 *** LD (strcd),HL
192 00BA' C3 *** JP CC12
193 00B9'
194 00BD' 21 *** LD HL,w0
195 00BC' C3 *** CALL smatch
196 00C1' 21 *** LD HL,w0+4
197 00C4' C4 *** CALL smatch
198 00C9' 20 2A JR NZ,C013
199 00CB'
200 00CB' C0 *** CALL h1hex
201 00CC' ES PUSH HL
202 00CE' FD 5B *** LD DE,(PTFRD)
203 00D0' F7 OR DE
204 00D4' FD 52 SBC HL,DE
205 00D6' E1 POP HL
206 00D9' DC *** CALL C_EXR14 ;illegal REG Error
207 00DB'
208 00DA' 22 *** LD (PTFRD),HL
209 00DD' 22 *** LD (PTFRD),HL
210 00ED'
211 00ED' D0 36 00 02 LD (IX),OTA
212 00EE' D0 23 INC IX
213 00EF' D0 75 00 LD (IX),I
214 00F0' D0 23 INC IX
215 00F2' D0 74 00 LD (IX),H
216 00F4' D0 23 INC IX
217 00F5'
218 00F6' 3A *** LD A,(sfgcd)
219 00F7' AT AND A
220 00FA' C2 *** JP NZ,C012
221 00FF' 3E 01 LD A,I
222 00F9' 32 *** LD (sfcd),A
223 00FC' 22 *** LD (strcd),HL
224 00FE' 22 *** LD (strw),HL
225 0102' C3 *** JP CC12
226 0105'
227 0105' 21 *** C013: LD HL,w-5
228 0106' C0 *** CALL smatch
229 0108' 21 *** LD HL,w-5+5
230 010A' C4 *** CALL smatch
231 0111' C2 *** JP NZ,C017
232 0114'
233 0114' D0 36 00 04 LD (IX),LIH
234 0116' D0 23 INC IX
235 011A' 21 *** LD HL,Rennance
236 011D' C3 *** CALL there
237 0120'
238 0120' 11 *** LD ,Rennance
239 0123' 21 *** LD HL,strLib
240 0126' C0 *** CALL C_finit

```

[illegible]

```

362 0231' LD A,1
363 0232' LD B,1
364 0233' LD C,1
365 0234' LD D,1
366 0235' LD E,1
367 0236' LD F,1
368 0237' LD H,1
369 0238' LD L,1
370 0239' LD A,1
371 0240' LD B,1
372 0241' LD C,1
373 0242' LD D,1
374 0243' LD E,1
375 0244' LD F,1
376 0245' LD H,1
377 0246' LD L,1
378 0247' LD A,1
379 0248' LD B,1
380 0249' LD C,1
381 0250' LD D,1
382 0251' LD E,1
383 0252' LD F,1
384 0253' LD H,1
385 0254' LD L,1
386 0255' LD A,1
387 0256' LD B,1
388 0257' LD C,1
389 0258' LD D,1
390 0259' LD E,1
391 0260' LD F,1
392 0261' LD H,1
393 0262' LD L,1
394 0263' LD A,1
395 0264' LD B,1
396 0265' LD C,1
397 0266' LD D,1
398 0267' LD E,1
399 0268' LD F,1
400 0269' LD H,1
401 0270' LD L,1
402 0271' LD A,1
403 0272' LD B,1
404 0273' LD C,1
405 0274' LD D,1
406 0275' LD E,1
407 0276' LD F,1
408 0277' LD H,1
409 0278' LD L,1
410 0279' LD A,1
411 0280' LD B,1
412 0281' LD C,1
413 0282' LD D,1
414 0283' LD E,1
415 0284' LD F,1
416 0285' LD H,1
417 0286' LD L,1
418 0287' LD A,1
419 0288' LD B,1
420 0289' LD C,1
421 0290' LD D,1
422 0291' LD E,1
423 0292' LD F,1
424 0293' LD H,1
425 0294' LD L,1
426 0295' LD A,1
427 0296' LD B,1
428 0297' LD C,1
429 0298' LD D,1
430 0299' LD E,1
431 0300' LD F,1
432 0301' LD H,1
433 0302' LD L,1
434 0303' LD A,1
435 0304' LD B,1
436 0305' LD C,1
437 0306' LD D,1
438 0307' LD E,1
439 0308' LD F,1
440 0309' LD H,1
441 0310' LD L,1
442 0311' LD A,1
443 0312' LD B,1
444 0313' LD C,1
445 0314' LD D,1
446 0315' LD E,1
447 0316' LD F,1
448 0317' LD H,1
449 0318' LD L,1
450 0319' LD A,1
451 0320' LD B,1
452 0321' LD C,1
453 0322' LD D,1
454 0323' LD E,1
455 0324' LD F,1
456 0325' LD H,1
457 0326' LD L,1
458 0327' LD A,1
459 0328' LD B,1
460 0329' LD C,1
461 0330' LD D,1
462 0331' LD E,1
463 0332' LD F,1
464 0333' LD H,1
465 0334' LD L,1
466 0335' LD A,1
467 0336' LD B,1
468 0337' LD C,1
469 0338' LD D,1
470 0339' LD E,1
471 0340' LD F,1
472 0341' LD H,1
473 0342' LD L,1
474 0343' LD A,1
475 0344' LD B,1
476 0345' LD C,1
477 0346' LD D,1
478 0347' LD E,1
479 0348' LD F,1
480 0349' LD H,1
481 0350' LD L,1
482 0351' LD A,1
483 0352' LD B,1
484 0353' LD C,1
485 0354' LD D,1
486 0355' LD E,1
487 0356' LD F,1
488 0357' LD H,1
489 0358' LD L,1
490 0359' LD A,1
491 0360' LD B,1
492 0361' LD C,1
493 0362' LD D,1
494 0363' LD E,1
495 0364' LD F,1
496 0365' LD H,1
497 0366' LD L,1
498 0367' LD A,1
499 0368' LD B,1
500 0369' LD C,1
501 0370' LD D,1
502 0371' LD E,1
503 0372' LD F,1
504 0373' LD H,1
505 0374' LD L,1
506 0375' LD A,1
507 0376' LD B,1
508 0377' LD C,1
509 0378' LD D,1
510 0379' LD E,1
511 0380' LD F,1
512 0381' LD H,1
513 0382' LD L,1
514 0383' LD A,1
515 0384' LD B,1
516 0385' LD C,1
517 0386' LD D,1
518 0387' LD E,1
519 0388' LD F,1
520 0389' LD H,1
521 0390' LD L,1
522 0391' LD A,1
523 0392' LD B,1
524 0393' LD C,1
525 0394' LD D,1
526 0395' LD E,1
527 0396' LD F,1
528 0397' LD H,1
529 0398' LD L,1
530 0399' LD A,1
531 0400' LD B,1
532 0401' LD C,1
533 0402' LD D,1
534 0403' LD E,1
535 0404' LD F,1
536 0405' LD H,1
537 0406' LD L,1
538 0407' LD A,1
539 0408' LD B,1
540 0409' LD C,1
541 0410' LD D,1
542 0411' LD E,1
543 0412' LD F,1
544 0413' LD H,1
545 0414' LD L,1
546 0415' LD A,1
547 0416' LD B,1
548 0417' LD C,1
549 0418' LD D,1
550 0419' LD E,1
551 0420' LD F,1
552 0421' LD H,1
553 0422' LD L,1
554 0423' LD A,1
555 0424' LD B,1
556 0425' LD C,1
557 0426' LD D,1
558 0427' LD E,1
559 0428' LD F,1
560 0429' LD H,1
561 0430' LD L,1
562 0431' LD A,1
563 0432' LD B,1
564 0433' LD C,1
565 0434' LD D,1
566 0435' LD E,1
567 0436' LD F,1
568 0437' LD H,1
569 0438' LD L,1
570 0439' LD A,1
571 0440' LD B,1
572 0441' LD C,1
573 0442' LD D,1
574 0443' LD E,1
575 0444' LD F,1
576 0445' LD H,1
577 0446' LD L,1
578 0447' LD A,1
579 0448' LD B,1
580 0449' LD C,1
581 0450' LD D,1
582 0451' LD E,1
583 0452' LD F,1
584 0453' LD H,1
585 0454' LD L,1
586 0455' LD A,1
587 0456' LD B,1
588 0457' LD C,1
589 0458' LD D,1
590 0459' LD E,1
591 0460' LD F,1
592 0461' LD H,1
593 0462' LD L,1
594 0463' LD A,1
595 0464' LD B,1
596 0465' LD C,1
597 0466' LD D,1
598 0467' LD E,1
599 0468' LD F,1
600 0469' LD H,1
601 0470' LD L,1
602 0471' LD A,1
603 0472' LD B,1
604 0473' LD C,1
605 0474' LD D,1
606 0475' LD E,1
607 0476' LD F,1
608 0477' LD H,1
609 0478' LD L,1
610 0479' LD A,1
611 0480' LD B,1
612 0481' LD C,1
613 0482' LD D,1
614 0483' LD E,1
615 0484' LD F,1
616 0485' LD H,1
617 0486' LD L,1
618 0487' LD A,1
619 0488' LD B,1
620 0489' LD C,1
621 0490' LD D,1
622 0491' LD E,1
623 0492' LD F,1
624 0493' LD H,1
625 0494' LD L,1
626 0495' LD A,1
627 0496' LD B,1
628 0497' LD C,1
629 0498' LD D,1
630 0499' LD E,1
631 0500' LD F,1
632 0501' LD H,1
633 0502' LD L,1
634 0503' LD A,1
635 0504' LD B,1
636 0505' LD C,1
637 0506' LD D,1
638 0507' LD E,1
639 0508' LD F,1
640 0509' LD H,1
641 0510' LD L,1
642 0511' LD A,1
643 0512' LD B,1
644 0513' LD C,1
645 0514' LD D,1
646 0515' LD E,1
647 0516' LD F,1
648 0517' LD H,1
649 0518' LD L,1
650 0519' LD A,1
651 0520' LD B,1
652 0521' LD C,1
653 0522' LD D,1
654 0523' LD E,1
655 0524' LD F,1
656 0525' LD H,1
657 0526' LD L,1
658 0527' LD A,1
659 0528' LD B,1
660 0529' LD C,1
661 0530' LD D,1
662 0531' LD E,1
663 0532' LD F,1
664 0533' LD H,1
665 0534' LD L,1
666 0535' LD A,1
667 0536' LD B,1
668 0537' LD C,1
669 0538' LD D,1
670 0539' LD E,1
671 0540' LD F,1
672 0541' LD H,1
673 0542' LD L,1
674 0543' LD A,1
675 0544' LD B,1
676 0545' LD C,1
677 0546' LD D,1
678 0547' LD E,1
679 0548' LD F,1
680 0549' LD H,1
681 0550' LD L,1
682 0551' LD A,1
683 0552' LD B,1
684 0553' LD C,1
685 0554' LD D,1
686 0555' LD E,1
687 0556' LD F,1
688 0557' LD H,1
689 0558' LD L,1
690 0559' LD A,1
691 0560' LD B,1
692 0561' LD C,1
693 0562' LD D,1
694 0563' LD E,1
695 0564' LD F,1
696 0565' LD H,1
697 0566' LD L,1
698 0567' LD A,1
699 0568' LD B,1
700 0569' LD C,1
701 0570' LD D,1
702 0571' LD E,1
703 0572' LD F,1
704 0573' LD H,1
705 0574' LD L,1
706 0575' LD A,1
707 0576' LD B,1
708 0577' LD C,1
709 0578' LD D,1
710 0579' LD E,1
711 0580' LD F,1
712 0581' LD H,1
713 0582' LD L,1
714 0583' LD A,1
715 0584' LD B,1
716 0585' LD C,1
717 0586' LD D,1
718 0587' LD E,1
719 0588' LD F,1
720 0589' LD H,1
721 0590' LD L,1
722 0591' LD A,1
723 0592' LD B,1
724 0593' LD C,1
725 0594' LD D,1
726 0595' LD E,1
727 0596' LD F,1
728 0597' LD H,1
729 0598' LD L,1
730 0599' LD A,1
731 0600' LD B,1
732 0601' LD C,1
733 0602' LD D,1
734 0603' LD E,1
735 0604' LD F,1
736 0605' LD H,1
737 0606' LD L,1
738 0607' LD A,1
739 0608' LD B,1
740 0609' LD C,1
741 0610' LD D,1
742 0611' LD E,1
743 0612' LD F,1
744 0613' LD H,1
745 0614' LD L,1
746 0615' LD A,1
747 0616' LD B,1
748 0617' LD C,1
749 0618' LD D,1
750 0619' LD E,1
751 0620' LD F,1
752 0621' LD H,1
753 0622' LD L,1
754 0623' LD A,1
755 0624' LD B,1
756 0625' LD C,1
757 0626' LD D,1
758 0627' LD E,1
759 0628' LD F,1
760 0629' LD H,1
761 0630' LD L,1
762 0631' LD A,1
763 0632' LD B,1
764 0633' LD C,1
765 0634' LD D,1
766 0635' LD E,1
767 0636' LD F,1
768 0637' LD H,1
769 0638' LD L,1
770 0639' LD A,1
771 0640' LD B,1
772 0641' LD C,1
773 0642' LD D,1
774 0643' LD E,1
775 0644' LD F,1
776 0645' LD H,1
777 0646' LD L,1
778 0647' LD A,1
779 0648' LD B,1
780 0649' LD C,1
781 0650' LD D,1
782 0651' LD E,1
783 0652' LD F,1
784 0653' LD H,1
785 0654' LD L,1
786 0655' LD A,1
787 0656' LD B,1
788 0657' LD C,1
789 0658' LD D,1
790 0659' LD E,1
791 0660' LD F,1
792 0661' LD H,1
793 0662' LD L,1
794 0663' LD A,1
795 0664' LD B,1
796 0665' LD C,1
797 0666' LD D,1
798 0667' LD E,1
799 0668' LD F,1
800 0669' LD H,1
801 0670' LD L,1
802 0671' LD A,1
803 0672' LD B,1
804 0673' LD C,1
805 0674' LD D,1
806 0675' LD E,1
807 0676' LD F,1
808 0677' LD H,1
809 0678' LD L,1
810 0679' LD A,1
811 0680' LD B,1
812 0681' LD C,1
813 0682' LD D,1
814 0683' LD E,1
815 0684' LD F,1
816 0685' LD H,1
817 0686' LD L,1
818 0687' LD A,1
819 0688' LD B,1
820 0689' LD C,1
821 0690' LD D,1
822 0691' LD E,1
823 0692' LD F,1
824 0693' LD H,1
825 0694' LD L,1
826 0695' LD A,1
827 0696' LD B,1
828 0697' LD C,1
829 0698' LD D,1
830 0699' LD E,1
831 0700' LD F,1
832 0701' LD H,1
833 0702' LD L,1
834 0703' LD A,1
835 0704' LD B,1
836 0705' LD C,1
837 0706' LD D,1
838 0707' LD E,1
839 0708' LD F,1
840 0709' LD H,1
841 0710' LD L,1
842 0711' LD A,1
843 0712' LD B,1
844 0713' LD C,1
845 0714' LD D,1
846 0715' LD E,1
847 0716' LD F,1
848 0717' LD H,1
849 0718' LD L,1
850 0719' LD A,1
851 0720' LD B,1
852 0721' LD C,1
853 0722' LD D,1
854 0723' LD E,1
855 0724' LD F,1
856 0725' LD H,1
857 0726' LD L,1
858 0727' LD A,1
859 0728' LD B,1
860 0729' LD C,1
861 0730' LD D,1
862 0731' LD E,1
863 0732' LD F,1
864 0733' LD H,1
865 0734' LD L,1
866 0735' LD A,1
867 0736' LD B,1
868 0737' LD C,1
869 0738' LD D,1
870 0739' LD E,1
871 0740' LD F,1
872 0741' LD H,1
873 0742' LD L,1
874 0743' LD A,1
875 0744' LD B,1
876 0745' LD C,1
877 0746' LD D,1
878 0747' LD E,1
879 0748' LD F,1
880 0749' LD H,1
881 0750' LD L,1
882 0751' LD A,1
883 0752' LD B,1
884 0753' LD C,1
885 0754' LD D,1
886 0755' LD E,1
887 0756' LD F,1
888 0757' LD H,1
889 0758' LD L,1
890 0759' LD A,1
891 0760' LD B,1
892 0761' LD C,1
893 0762' LD D,1
894 0763' LD E,1
895 0764' LD F,1
896 0765' LD H,1
897 0766' LD L,1
898 0767' LD A,1
899 0768' LD B,1
900 0769' LD C,1
901 0770' LD D,1
902 0771' LD E,1
903 0772' LD F,1
904 0773' LD H,1
905 0774' LD L,1
906 0775' LD A,1
907 0776' LD B,1
908 0777' LD C,1
909 0778' LD D,1
910 0779' LD E,1
911 0780' LD F,1
912 0781' LD H,1
913 0782' LD L,1
914 0783' LD A,1
915 0784' LD B,1
916 0785' LD C,1
917 0786' LD D,1
918 0787' LD E,1
919 0788' LD F,1
920 0789' LD H,1
921 0790' LD L,1
922 0791' LD A,1
923 0792' LD B,1
924 0793' LD C,1
925 0794' LD D,1
926 0795' LD E,1
927 0796' LD F,1
928 0797' LD H,1
929 0798' LD L,1
930 0799' LD A,1
931 0800' LD B,1
932 0801' LD C,1
933 0802' LD D,1
934 0803' LD E,1
935 0804' LD F,1
936 0805' LD H,1
937 0806' LD L,1
938 0807' LD A,1
939 0808' LD B,1
940 0809' LD C,1
941 0810' LD D,1
942 0811' LD E,1
943 0812' LD F,1
944 0813' LD H,1
945 0814' LD L,1
946 0815' LD A,1
947 0816' LD B,1
948 0817' LD C,1
949 0818' LD D,1
950 0819' LD E,1
951 0820' LD F,1
952 0821' LD H,1
953 0822' LD L,1
954 0823' LD A,1
955 0824' LD B,1
956 0825' LD C,1
957 0826' LD D,1
958 0827' LD E,1
959 0828' LD F,1
960 0829' LD H,1
961 0830' LD L,1
962 0831' LD A,1
963 0832' LD B,1
964 0833' LD C,1
965 0834' LD D,1
966 0835' LD E,1
967 0836' LD F,1
968 0837' LD H,1
969 0838' LD L,1
970 0839' LD A,1
971 0840' LD B,1
972 0841' LD C,1
973 0842' LD D,1
974 0843' LD E,1
975 0844' LD F,1
976 0845' LD H,1
977 0846' LD L,1
978 0847' LD A,1
979 0848' LD B,1
980 0849' LD C,1
981 0850' LD D,1
982 0851' LD E,1
983 0852' LD F,1
984 0853' LD H,1
985 0854' LD L,1
986 0855' LD A,1
987 0856' LD B,1
988 0857' LD C,1
989 0858' LD D,1
990 0859' LD E,1
991 0860' LD F,1
992 0861' LD H,1
993 0862' LD L,1
994 0863' LD A,1
995 0864' LD B,1
996 0865' LD C,1
997 0866' LD D,1
998 0867' LD E,1
999 0868' LD F,1
1000 0869' LD H,1

```



```

846 0628' 41 53 53 73
847 0629' 72 28 45 72 72
848 0630' 67 72 80 00
849 0631'
850 0632' 21 *** (Ferr: LD HL,CC55
851 0633' CD *** CALL _puts
852 0634' C9 RET
853 0635' 49 6C 6C 65 67 C0B5:DB 'Illegal LIB File Error',CR,0
854 0636' 61 6C 28 4C 48
855 0637' 42 28 48 65 6C
856 0638' 65 28 45 72 72
857 0639' 67 72 80 00
858 0640'
859 0641' 21 *** ER14: LD HL,STR14
860 0642' CD *** CALL _puts
861 0643' C9 RET
862 0644' 49 6C 6C 65 67 STR14:DB 'Illegal ORG Error',CR,0
863 0645' 61 6C 28 4C 48
864 0646' 42 28 48 65 6C
865 0647' 65 28 45 72 72
866 0648' 67 72 80 00
867 0649'
868 0650' 3A *** exit: LD A,(pass)
869 0651' A7 AND A
870 0652' C4 *** CALL N2_CLOSE
871 0653' C7 FA 17 JP _NOT
872 0654'
873 0655' 70 25 match: PUSH DE
874 0656' 7E POP DE ;LD DE,IV
875 0657' A7 AND A
876 0658' 7E POP DE ;LD DE,IV
877 0659' 7E POP DE ;LD DE,IV
878 0660' 7E POP DE ;LD DE,IV
879 0661' 7E POP DE ;LD DE,IV
880 0662' 7E POP DE ;LD DE,IV
881 0663' 7E POP DE ;LD DE,IV
882 0664' 7E POP DE ;LD DE,IV
883 0665' 7E POP DE ;LD DE,IV
884 0666' 7E POP DE ;LD DE,IV
885 0667' 7E POP DE ;LD DE,IV
886 0668' 7E POP DE ;LD DE,IV
887 0669' 7E POP DE ;LD DE,IV
888 0670' 7E POP DE ;LD DE,IV
889 0671' 7E POP DE ;LD DE,IV
890 0672' 7E POP DE ;LD DE,IV
891 0673' 7E POP DE ;LD DE,IV
892 0674' 7E POP DE ;LD DE,IV
893 0675' 7E POP DE ;LD DE,IV
894 0676' 7E POP DE ;LD DE,IV
895 0677' 7E POP DE ;LD DE,IV
896 0678' 7E POP DE ;LD DE,IV
897 0679' 7E POP DE ;LD DE,IV
898 0680' 7E POP DE ;LD DE,IV
899 0681' 7E POP DE ;LD DE,IV
900 0682' 7E POP DE ;LD DE,IV
901 0683' 7E POP DE ;LD DE,IV
902 0684' 7E POP DE ;LD DE,IV
903 0685' 7E POP DE ;LD DE,IV
904 0686' 7E POP DE ;LD DE,IV
905 0687' 7E POP DE ;LD DE,IV
906 0688' 7E POP DE ;LD DE,IV
907 0689' 7E POP DE ;LD DE,IV
908 0690' 7E POP DE ;LD DE,IV
909 0691' 7E POP DE ;LD DE,IV
910 0692' 7E POP DE ;LD DE,IV
911 0693' 7E POP DE ;LD DE,IV
912 0694' 7E POP DE ;LD DE,IV
913 0695' 7E POP DE ;LD DE,IV
914 0696' 7E POP DE ;LD DE,IV
915 0697' 7E POP DE ;LD DE,IV

```

```

916 0698'
917 0699' 14 fcopy: LD B,LENFL
918 0700'
919 0701' XOR A
920 0702' LD (DE),A
921 0703'
922 0704' LD A,(HL)
923 0705' INC HL
924 0706' CALL isfchr
925 0707' JR C,CC105
926 0708' DEC B
927 0709' JR Z,CC105
928 0710'
929 0711' LD (DE),A
930 0712' INC DE
931 0713' XOR A
932 0714' LD (DE),A
933 0715' JR CC104
934 0716'
935 0717' INC HL
936 0718' LD (DE),HL
937 0719' RET
938 0720'
939 0721' XOR A
940 0722' LD (DE),A
941 0723'
942 0724' LD A,(HL)
943 0725' INC HL
944 0726' CALL isfchr
945 0727' JR C,CC105
946 0728' DEC B
947 0729' JR Z,CC105
948 0730'
949 0731' LD (DE),A
950 0732' INC DE
951 0733' XOR A
952 0734' LD (DE),A
953 0735' JR CC104
954 0736'
955 0737' error: CALL _puts
956 0738' JP _exit
957 0739'
958 0740' fcopy2: LD B,LENFL
959 0741'
960 0742' LD (DE),A
961 0743'
962 0744' LD A,(HL)
963 0745' INC HL
964 0746' CALL isfchr
965 0747' JR C,CC105
966 0748' DEC B
967 0749' JR Z,CC105
968 0750'
969 0751' LD (DE),A
970 0752' INC DE
971 0753' XOR A
972 0754' LD (DE),A
973 0755' JR CC104
974 0756'
975 0757' INC HL
976 0758' LD (DE),HL
977 0759' RET
978 0760'
979 0761' XOR A
980 0762' LD (DE),A
981 0763'
982 0764' LD A,(HL)
983 0765' INC HL
984 0766' CALL isfchr
985 0767' JR C,CC105
986 0768' DEC B
987 0769' JR Z,CC105
988 0770'
989 0771' LD (DE),A
990 0772' INC DE
991 0773' XOR A
992 0774' LD (DE),A
993 0775' JR CC104
994 0776'
995 0777' INC HL
996 0778' LD (DE),HL
997 0779' RET
998 0780'
999 0781' XOR A
1000 0782' LD (DE),A

```

```

1001 0783'
1002 0784' RET NC
1003 0785'
1004 0786' CALL ADD_HL_HL
1005 0787' ADD HL,HL
1006 0788' ADD HL,HL
1007 0789' ADD HL,HL
1008 0790' ADD HL,HL
1009 0791' ADD HL,HL
1010 0792' ADD HL,HL
1011 0793' ADD HL,HL
1012 0794' ADD HL,HL
1013 0795' ADD HL,HL
1014 0796' ADD HL,HL
1015 0797' ADD HL,HL
1016 0798' ADD HL,HL
1017 0799' ADD HL,HL
1018 0800' ADD HL,HL
1019 0801' ADD HL,HL
1020 0802' ADD HL,HL
1021 0803' ADD HL,HL
1022 0804' ADD HL,HL
1023 0805' ADD HL,HL
1024 0806' ADD HL,HL
1025 0807' ADD HL,HL
1026 0808' ADD HL,HL
1027 0809' ADD HL,HL
1028 0810' ADD HL,HL
1029 0811' ADD HL,HL
1030 0812' ADD HL,HL
1031 0813' ADD HL,HL
1032 0814' ADD HL,HL
1033 0815' ADD HL,HL
1034 0816' ADD HL,HL
1035 0817' ADD HL,HL
1036 0818' ADD HL,HL
1037 0819' ADD HL,HL
1038 0820' ADD HL,HL
1039 0821' ADD HL,HL
1040 0822' ADD HL,HL
1041 0823' ADD HL,HL
1042 0824' ADD HL,HL
1043 0825' ADD HL,HL
1044 0826' ADD HL,HL
1045 0827' ADD HL,HL
1046 0828' ADD HL,HL
1047 0829' ADD HL,HL
1048 0830' ADD HL,HL
1049 0831' ADD HL,HL
1050 0832' ADD HL,HL
1051 0833' ADD HL,HL
1052 0834' ADD HL,HL
1053 0835' ADD HL,HL
1054 0836' ADD HL,HL
1055 0837' ADD HL,HL
1056 0838' ADD HL,HL
1057 0839' ADD HL,HL
1058 0840' ADD HL,HL
1059 0841' ADD HL,HL
1060 0842' ADD HL,HL
1061 0843' ADD HL,HL
1062 0844' ADD HL,HL
1063 0845' ADD HL,HL
1064 0846' ADD HL,HL
1065 0847' ADD HL,HL
1066 0848' ADD HL,HL
1067 0849' ADD HL,HL
1068 0850' ADD HL,HL
1069 0851' ADD HL,HL
1070 0852' ADD HL,HL
1071 0853' ADD HL,HL
1072 0854' ADD HL,HL
1073 0855' ADD HL,HL
1074 0856' ADD HL,HL
1075 0857' ADD HL,HL
1076 0858' ADD HL,HL
1077 0859' ADD HL,HL
1078 0860' ADD HL,HL
1079 0861' ADD HL,HL
1080 0862' ADD HL,HL
1081 0863' ADD HL,HL
1082 0864' ADD HL,HL
1083 0865' ADD HL,HL
1084 0866' ADD HL,HL
1085 0867' ADD HL,HL
1086 0868' ADD HL,HL
1087 0869' ADD HL,HL
1088 0870' ADD HL,HL
1089 0871' ADD HL,HL
1090 0872' ADD HL,HL
1091 0873' ADD HL,HL
1092 0874' ADD HL,HL
1093 0875' ADD HL,HL
1094 0876' ADD HL,HL
1095 0877' ADD HL,HL
1096 0878' ADD HL,HL
1097 0879' ADD HL,HL
1098 0880' ADD HL,HL
1099 0881' ADD HL,HL
1100 0882' ADD HL,HL

```

リストB

```

1 0000'
2 0001'
3 0002'
4 0003'
5 0004'
6 0005'
7 0006'
8 0007'
9 0008'
10 0009'
11 0010'
12 0011'
13 0012'
14 0013'
15 0014'
16 0015'
17 0016'
18 0017'
19 0018'
20 0019'
21 0020'
22 0021'
23 0022'
24 0023'
25 0024'
26 0025'
27 0026'
28 0027'
29 0028'
30 0029'
31 0030'
32 0031'
33 0032'
34 0033'
35 0034'
36 0035'
37 0036'
38 0037'
39 0038'
40 0039'
41 0040'
42 0041'
43 0042'
44 0043'
45 0044'
46 0045'
47 0046'
48 0047'
49 0048'
50 0049'
51 0050'
52 0051'
53 0052'
54 0053'
55 0054'
56 0055'
57 0056'
58 0057'
59 0058'
60 0059'
61 0060'
62 0061'
63 0062'
64 0063'
65 0064'
66 0065'
67 0066'
68 0067'
69 0068'
70 0069'
71 0070'
72 0071'
73 0072'
74 0073'
75 0074'
76 0075'
77 0076'
78 0077'
79 0078'
80 0079'
81 0080'
82 0081'
83 0082'
84 0083'
85 0084'
86 0085'
87 0086'
88 0087'
89 0088'
90 0089'
91 0090'
92 0091'
93 0092'
94 0093'
95 0094'
96 0095'
97 0096'
98 0097'
99 0098'
100 0099'

```

```

84 0000' ED 73 ** LD (SP+0),SP
85 0001' LD A,(ITEM)
86 0002' AND A
87 0003' JP Z,ITEM0
88 0004' CP 20H
89 0005' JP Z,ITEM1
90 0006' CP 60H
91 0007' JP Z,ITEM2
92 0008' LD A,(ITEM)
93 0009' AND 0FH
94 0010' CP 90H
95 0011' JP Z,ITEM3
96 0012' CP 30H
97 0013' JP Z,ITEM4
98 0014' CP 00H
99 0015' JP Z,ITEM5
100 0016' LD A,(ITEM)
101 0017' AND 0FH
102 0018' CP 90H
103 0019' JP Z,ITEM6
104 0020' LD A,(ITEM)
105 0021' AND 0FH
106 0022' CP 90H
107 0023' JP Z,ITEM7
108 0024' LD A,(ITEM)
109 0025' AND 0FH
110 0026' CP 90H
111 0027' JP Z,ITEM8
112 0028' LD A,(ITEM)
113 0029' AND 0FH
114 0030' CP 90H
115 0031' JP Z,ITEM9
116 0032' LD A,(ITEM)
117 0033' AND 0FH
118 0034' CP 90H
119 0035' JP Z,ITEM10
120 0036' LD A,(ITEM)
121 0037' AND 0FH
122 0038' CP 90H
123 0039' JP Z,ITEM11
124 0040' LD A,(ITEM)
125 0041' AND 0FH
126 0042' CP 90H
127 0043' JP Z,ITEM12
128 0044' LD A,(ITEM)
129 0045' AND 0FH
130 0046' CP 90H
131 0047' JP Z,ITEM13
132 0048' LD A,(ITEM)
133 0049' AND 0FH
134 0050' CP 90H
135 0051' JP Z,ITEM14
136 0052' LD A,(ITEM)
137 0053' AND 0FH
138 0054' CP 90H
139 0055' JP Z,ITEM15
140 0056' LD A,(ITEM)
141 0057' AND 0FH
142 0058' CP 90H
143 0059' JP Z,ITEM16
144 0060' LD A,(ITEM)
145 0061' AND 0FH
146 0062' CP 90H
147 0063' JP Z,ITEM17
148 0064' LD A,(ITEM)
149 0065' AND 0FH
150 0066' CP 90H
151 0067' JP Z,ITEM18
152 0068' LD A,(ITEM)
153 0069' AND 0FH
154 0070' CP 90H
155 0071' JP Z,ITEM19
156 0072' LD A,(ITEM)
157 0073' AND 0FH
158 0074' CP 90H
159 0075' JP Z,ITEM20
160 0076' LD A,(ITEM)
161 0077' AND 0FH
162 0078' CP 90H
163 0079' JP Z,ITEM21
164 0080' LD A,(ITEM)
165 0081' AND 0FH
166 0082' CP 90H
167 0083' JP Z,ITEM22
168 0084' LD A,(ITEM)
169 0085' AND 0FH
170 0086' CP 90H
171 0087' JP Z,ITEM23
172 0088' LD A,(ITEM)
173 0089' AND 0FH
174 0090' CP 90H
175 0091' JP Z,ITEM24
176 0092' LD A,(ITEM)
177 0093' AND 0FH
178 0094' CP 90H
179 0095' JP Z,ITEM25
180 0096' LD A,(ITEM)
181 0097' AND 0FH
182 0098' CP 90H
183 0099' JP Z,ITEM26
184 0100' LD A,(ITEM)
185 0101' AND 0FH
186 0102' CP 90H
187 0103' JP Z,ITEM27
188 0104' LD A,(ITEM)
189 0105' AND 0FH
190 0106' CP 90H
191 0107' JP Z,ITEM28
192 0108' LD A,(ITEM)
193 0109' AND 0FH
194 0110' CP 90H
195 0111' JP Z,ITEM29
196 0112' LD A,(ITEM)
197 0113' AND 0FH
198 0114' CP 90H
199 0115' JP Z,ITEM30
200 0116' LD A,(ITEM)
201 0117' AND 0FH
202 0118' CP 90H
203 0119' JP Z,ITEM31
204 0120' LD A,(ITEM)
205 0121' AND 0FH
206 0122' CP 90H
207 0123' JP Z,ITEM32
208 0124' LD A,(ITEM)
209 0125' AND 0FH
210 0126' CP 90H
211 0127' JP Z,ITEM33
212 0128' LD A,(ITEM)
213 0129' AND 0FH
214 0130' CP 90H
215 0131' JP Z,ITEM34
216 0132' LD A,(ITEM)
217 0133' AND 0FH
218 0134' CP 90H
219 0135' JP Z,ITEM35
220 0136' LD A,(ITEM)
221 0137' AND 0FH
222 0138' CP 90H
223 0139' JP Z,ITEM36
224 0140' LD A,(ITEM)
225 0141' AND 0FH
226 0142' CP 90H
227 0143' JP Z,ITEM37
228 0144' LD A,(ITEM)
229 0145' AND 0FH
230 0146' CP 90H
231 0147' JP Z,ITEM38
232 0148' LD A,(ITEM)
233 0149' AND 0FH
234 0150' CP 90H
235 0151' JP Z,ITEM39
236 0152' LD A,(ITEM)
237 0153' AND 0FH
238 0154' CP 90H
239 0155' JP Z,ITEM40
240 0156' LD A,(ITEM)
241 0157' AND 0FH
242 0158' CP 90H
243 0159' JP Z,ITEM41
244 0160' LD A,(ITEM)
245 0161' AND 0FH
246 0162' CP 90H
247 0163' JP Z,ITEM42
248 0164' LD A,(ITEM)
249 0165' AND 0FH
250 0166' CP 90H
251 0167' JP Z,ITEM43
252 0168' LD A,(ITEM)
253 0169' AND 0FH
254 0170' CP 90H
255 0171' JP Z,ITEM44
256 0172' LD A,(ITEM)
257 0173' AND 0FH
258 0174' CP 90H
259 0175' JP Z,ITEM45
260 0176' LD A,(ITEM)
261 0177' AND 0FH
262 0178' CP 90H
263 0179' JP Z,ITEM46
264 0180' LD A,(ITEM)
265 0181' AND 0FH
266 0182' CP 90H
267 0183' JP Z,ITEM47
268 0184' LD A,(ITEM)
269 0185' AND 0FH
270 0186' CP 90H
271 0187' JP Z,ITEM48
272 0188' LD A,(ITEM)
273 0189' AND 0FH
274 0190' CP 90H
275 0191' JP Z,ITEM49
276 0192' LD A,(ITEM)
277 0193' AND 0FH
278 0194' CP 90H
279 0195' JP Z,ITEM50
280 0196' LD A,(ITEM)
281 0197' AND 0FH
282 0198' CP 90H
283 0199' JP Z,ITEM51
284 0200' LD A,(ITEM)
285 0201' AND 0FH
286 0202' CP 90H
287 0203' JP Z,ITEM52
288 0204' LD A,(ITEM)
289 0205' AND 0FH
290 0206' CP 90H
291 0207' JP Z,ITEM53
292 0208' LD A,(ITEM)
293 0209' AND 0FH
294 0210' CP 90H
295 0211' JP Z,ITEM54
296 0212' LD A,(ITEM)
297 0213' AND 0FH
298 0214' CP 90H
299 0215' JP Z,ITEM55
300 0216' LD A,(ITEM)
301 0217' AND 0FH
302 0218' CP 90H
303 0219' JP Z,ITEM56
304 0220' LD A,(ITEM)
305 0221' AND 0FH
306 0222' CP 90H
307 0223' JP Z,ITEM57
308 0224' LD A,(ITEM)
309 0225' AND 0FH
310 0226' CP 90H
311 0227' JP Z,ITEM58
312 0228' LD A,(ITEM)
313 0229' AND 0FH
314 0230' CP 90H
315 0231' JP Z,ITEM59
316 0232' LD A,(ITEM)
317 0233' AND 0FH
318 0234' CP 90H
319 0235' JP Z,ITEM60
320 0236' LD A,(ITEM)
321 0237' AND 0FH
322 0238' CP 90H
323 0239' JP Z,ITEM61
324 0240' LD A,(ITEM)
325 0241' AND 0FH
326 0242' CP 90H
327 0243' JP Z,ITEM62
328 0244' LD A,(ITEM)
329 0245' AND 0FH
330 0246' CP 90H
331 0247' JP Z,ITEM63
332 0248' LD A,(ITEM)
333 0249' AND 0FH
334 0250' CP 90H
335 0251' JP Z,ITEM64
336 0252' LD A,(ITEM)
337 0253' AND 0FH
338 0254' CP 90H
339 0255' JP Z,ITEM65
340 0256' LD A,(ITEM)
341 0257' AND 0FH
342 0258' CP 90H
343 0259' JP Z,ITEM66
344 0260' LD A,(ITEM)
345 0261' AND 0FH
346 0262' CP 90H
347 0263' JP Z,ITEM67
348 0264' LD A,(ITEM)
349 0265' AND 0FH
350 0266' CP 90H
351 0267' JP Z,ITEM68
352 0268' LD A,(ITEM)
353 0269' AND 0FH
354 0270' CP 90H
355 0271' JP Z,ITEM69
356 0272' LD A,(ITEM)
357 0273' AND 0FH
358 0274' CP 90H
359 0275' JP Z,ITEM70
360 0276' LD A,(ITEM)
361 0277' AND 0FH
362 0278' CP 90H
363 0279' JP Z,ITEM71
364 0280' LD A,(ITEM)
365 0281' AND 0FH
366 0282' CP 90H
367 0283' JP Z,ITEM72
368 0284' LD A,(ITEM)
369 0285' AND 0FH
370 0286' CP 90H
371 0287' JP Z,ITEM73
372 0288' LD A,(ITEM)
373 0289' AND 0FH
374 0290' CP 90H
375 0291' JP Z,ITEM74
376 0292' LD A,(ITEM)
377 0293' AND 0FH
378 0294' CP 90H
379 0295' JP Z,ITEM75
380 0296' LD A,(ITEM)
381 0297' AND 0FH
382 0298' CP 90H
383 0299' JP Z,ITEM76
384 0300' LD A,(ITEM)
385 0301' AND 0FH
386 0302' CP 90H
387 0303' JP Z,ITEM77
388 0304' LD A,(ITEM)
389 0305' AND 0FH
390 0306' CP 90H
391 0307' JP Z,ITEM78
392 0308' LD A,(ITEM)
393 0309' AND 0FH
394 0310' CP 90H
395 0311' JP Z,ITEM79
396 0312' LD A,(ITEM)
397 0313' AND 0FH
398 0314' CP 90H
399 0315' JP Z,ITEM80
400 0316' LD A,(ITEM)
401 0317' AND 0FH
402 0318' CP 90H
403 0319' JP Z,ITEM81
404 0320' LD A,(ITEM)
405 0321' AND 0FH
406 0322' CP 90H
407 0323' JP Z,ITEM82
408 0324' LD A,(ITEM)
409 0325' AND 0FH
410 0326' CP 90H
411 0327' JP Z,ITEM83
412 0328' LD A,(ITEM)
413 0329' AND 0FH
414 0330' CP 90H
415 0331' JP Z,ITEM84
416 0332' LD A,(ITEM)
417 0333' AND 0FH
418 0334' CP 90H
419 0335' JP Z,ITEM85
420 0336' LD A,(ITEM)
421 0337' AND 0FH
422 0338' CP 90H
423 0339' JP Z,ITEM86
424 0340' LD A,(ITEM)
425 0341' AND 0FH
426 0342' CP 90H
427 0343' JP Z,ITEM87
428 0344' LD A,(ITEM)
429 0345' AND 0FH
430 0346' CP 90H
431 0347' JP Z,ITEM88
432 0348' LD A,(ITEM)
433 0349' AND 0FH
434 0350' CP 90H
435 0351' JP Z,ITEM89
436 0352' LD A,(ITEM)
437 0353' AND 0FH
438 0354' CP 90H
439 0355' JP Z,ITEM90
440 0356' LD A,(ITEM)
441 0357' AND 0FH
442 0358' CP 90H
443 0359' JP Z,ITEM91
444 0360' LD A,(ITEM)
445 0361' AND 0FH
446 0362' CP 90H
447 0363' JP Z,ITEM92
448 0364' LD A,(ITEM)
449 0365' AND 0FH
450 0366' CP 90H
451 0367' JP Z,ITEM93
452 0368' LD A,(ITEM)
453 0369' AND 0FH
454 0370' CP 90H
455 0371' JP Z,ITEM94
456 0372' LD A,(ITEM)
457 0373' AND 0FH
458 0374' CP 90H
459 0375' JP Z,ITEM95
460 0376' LD A,(ITEM)
461 0377' AND 0FH
462 0378' CP 90H
463 0379' JP Z,ITEM96
464 0380' LD A,(ITEM)
465 0381' AND 0FH
466 0382' CP 90H
467 0383' JP Z,ITEM97
468 0384' LD A,(ITEM)
469 0385' AND 0FH
470 0386' CP 90H
471 0387' JP Z,ITEM98
472 0388' LD A,(ITEM)
473 0389' AND 0FH
474 0390' CP 90H
475 0391' JP Z,ITEM99
476 0392' LD A,(ITEM)
477 0393' AND 0FH
478 0394' CP 90H
479 0395' JP Z,ITEM100
480 0396' LD A,(ITEM)
481 0397' AND 0FH
482 0398' CP 90H
483 0399' JP Z,ITEM101
484 0400' LD A,(ITEM)
485 0401' AND 0FH
486 0402' CP 90H
487 0403' JP Z,ITEM102
488 0404' LD A,(ITEM)
489 0405' AND 0FH
490 0406' CP 90H
491 0407' JP Z,ITEM103
492 0408' LD A,(ITEM)
493 0409' AND 0FH
494 0410' CP 90H
495 0411' JP Z,ITEM104
496 0412' LD A,(ITEM)
497 0413' AND 0FH
498 0414' CP 90H
499 0415' JP Z,ITEM105
500 0416' LD A,(ITEM)
501 0417' AND 0FH
502 0418' CP 90H
503 0419' JP Z,ITEM106
504 0420' LD A,(ITEM)
505 0421' AND 0FH
506 0422' CP 90H
507 0423' JP Z,ITEM107
508 0424' LD A,(ITEM)
509 0425' AND 0FH
510 0426' CP 90H
511 0427' JP Z,ITEM108
512 0428' LD A,(ITEM)
513 0429' AND 0FH
514 0430' CP 90H
515 0431' JP Z,ITEM109
516 0432' LD A,(ITEM)
517 0433' AND 0FH
518 0434' CP 90H
519 0435' JP Z,ITEM110
520 0436' LD A,(ITEM)
521 0437' AND 0FH
522 0438' CP 90H
523 0439' JP Z,ITEM111
524 0440' LD A,(ITEM)
525 0441' AND 0FH
526 0442' CP 90H
527 0443' JP Z,ITEM112
528 0444' LD A,(ITEM)
529 0445' AND 0FH
530 0446' CP 90H
531 0447' JP Z,ITEM113
532 0448' LD A,(ITEM)
533 0449' AND 0FH
534 0450' CP 90H
535 0451' JP Z,ITEM114
536 0452' LD A,(ITEM)
537 0453' AND 0FH
538 0454' CP 90H
539 0455' JP Z,ITEM115
540 0456' LD A,(ITEM)
541 0457' AND 0FH
542 0458' CP 90H
543 0459' JP Z,ITEM116
544 0460' LD A,(ITEM)
545 0461' AND 0FH
546 0462' CP 90H
547 0463' JP Z,ITEM117
548 0464' LD A,(ITEM)
549 0465' AND 0FH
550 0466' CP 90H
551 0467' JP Z,ITEM118
552 0468' LD A,(ITEM)
553 0469' AND 0FH
554 0470' CP 90H
555 0471' JP Z,ITEM119
556 0472' LD A,(ITEM)
557 0473' AND 0FH
558 0474' CP 90H
559 0475' JP Z,ITEM120
560 0476' LD A,(ITEM)
561 0477' AND 0FH
562 0478' CP 90H
563 0479' JP Z,ITEM121
564 0480' LD A,(ITEM)
565 0481' AND 0FH
566 0482' CP 90H
```



```

727 044B' 70      LD      A,B
730 044C' 01      OR      CR
733 044D' 2F F7   JR      NZ,DEF1
740 044F'          ; Chain last struct with new struct
741 044F' 20      BCF      HL
743 0449' 20      SBC      HL
744 04B1' ED 45 *** LD      BC,(LBIPTX)
745 04B5' CD ***   CALL POKE_BC
746 04B7'          ; ** Make Struct of Labels **
747 04B8' 2A ***   LD      HL,(LBIPTX)
749 04B0' 01 00   LD      BC,0
750 04B5' CD ***   CALL POKE_BC
751 04C1' 3A ***   LD      A,(SV_FIG)
752 04C4' CD ***   CALL POKE_I
753 04C7' CD 40 *** LD      BC,(SV_VAR)
754 04C8' CD ***   CALL POKE_BC
755 04C9'          LD      DE,LBIPTX
756 04CE' 11 ***   DEF2: LD      A,(DE)
757 04D1' TA      INC      DE
758 04D2' 10      INC      DE
759 04D3' F5      PUSH     AF
760 04D4' CD ***   CALL POKE_I
761 04D7' 01      POP      AF
762 04D8' 07      JR      NZ,DEF2
763 04D9' 20 F6   JR      NZ,DEF2
764 04DB'          LD      (LBIPTX),HL
765 04DE' 22 ***   LD      HL,A
766 04E0'          RET
767 04E1'          LD      HL,(SV_STR)
768 04E2' 23      INC      HL
769 04E3' 23      INC      HL
770 04E4' 3A ***   LD      A,(SV_FIG)
771 04E5' CD ***   CALL POKE_I
772 04E6' ED 45 *** LD      BC,(SV_VAR)
773 04E7' CD ***   CALL POKE_BC
774 04E8' CD ***   LD      BC,(SV_VAR)
775 04E9' CD ***   CALL POKE_BC
776 04FA' 07      JR      NZ,DEF2
777 04FB'          LD      HL,A
778 04FC'          LD      HL,(SV_STR)
779 04FD' 23      INC      HL
780 04FE'          LD      HL,A
781 04FF'          LD      HL,A
782 0500'          LD      HL,A
783 0501'          LD      HL,A
784 0502'          LD      HL,A
785 0503'          LD      HL,A
786 0504'          LD      HL,A
787 0505'          LD      HL,A
788 0506'          LD      HL,A
789 0507'          LD      HL,A
790 0508'          LD      HL,A
791 0509'          LD      HL,A
792 0510'          LD      HL,A
793 0511'          LD      HL,A
794 0512'          LD      HL,A
795 0513'          LD      HL,A
796 0514'          LD      HL,A
797 0515'          LD      HL,A
798 0516'          LD      HL,A
799 0517'          LD      HL,A
800 0518'          LD      HL,A
801 0519'          LD      HL,A
802 0520'          LD      HL,A
803 0521'          LD      HL,A
804 0522'          LD      HL,A
805 0523'          LD      HL,A
806 0524'          LD      HL,A
807 0525'          LD      HL,A
808 0526'          LD      HL,A
809 0527'          LD      HL,A
810 0528'          LD      HL,A
811 0529'          LD      HL,A
812 0530'          LD      HL,A
813 0531'          LD      HL,A
814 0532'          LD      HL,A
815 0533'          LD      HL,A
816 0534'          LD      HL,A
817 0535'          LD      HL,A
818 0536'          LD      HL,A
819 0537'          LD      HL,A
820 0538'          LD      HL,A
821 0539'          LD      HL,A
822 0540'          LD      HL,A
823 0541'          LD      HL,A
824 0542'          LD      HL,A
825 0543'          LD      HL,A
826 0544'          LD      HL,A
827 0545'          LD      HL,A
828 0546'          LD      HL,A
829 0547'          LD      HL,A
830 0548'          LD      HL,A
831 0549'          LD      HL,A
832 0550'          LD      HL,A
833 0551'          LD      HL,A
834 0552'          LD      HL,A
835 0553'          LD      HL,A
836 0554'          LD      HL,A
837 0555'          LD      HL,A
838 0556'          LD      HL,A
839 0557'          LD      HL,A
840 0558'          LD      HL,A
841 0559'          LD      HL,A
842 0560'          LD      HL,A
843 0561'          LD      HL,A
844 0562'          LD      HL,A
845 0563'          LD      HL,A
846 0564'          LD      HL,A
847 0565'          LD      HL,A
848 0566'          LD      HL,A
849 0567'          LD      HL,A
850 0568'          LD      HL,A
851 0569'          LD      HL,A
852 0570'          LD      HL,A
853 0571'          LD      HL,A
854 0572'          LD      HL,A
855 0573'          LD      HL,A
856 0574'          LD      HL,A
857 0575'          LD      HL,A
858 0576'          LD      HL,A
859 0577'          LD      HL,A
860 0578'          LD      HL,A
861 0579'          LD      HL,A
862 0580'          LD      HL,A
863 0581'          LD      HL,A
864 0582'          LD      HL,A
865 0583'          LD      HL,A
866 0584'          LD      HL,A
867 0585'          LD      HL,A
868 0586'          LD      HL,A
869 0587'          LD      HL,A
870 0588'          LD      HL,A
871 0589'          LD      HL,A
872 0590'          LD      HL,A
873 0591'          LD      HL,A
874 0592'          LD      HL,A
875 0593'          LD      HL,A
876 0594'          LD      HL,A
877 0595'          LD      HL,A
878 0596'          LD      HL,A
879 0597'          LD      HL,A
880 0598'          LD      HL,A
881 0599'          LD      HL,A
882 0600'          LD      HL,A
883 0601'          LD      HL,A
884 0602'          LD      HL,A
885 0603'          LD      HL,A
886 0604'          LD      HL,A
887 0605'          LD      HL,A
888 0606'          LD      HL,A
889 0607'          LD      HL,A
890 0608'          LD      HL,A
891 0609'          LD      HL,A
892 0610'          LD      HL,A
893 0611'          LD      HL,A
894 0612'          LD      HL,A
895 0613'          LD      HL,A
896 0614'          LD      HL,A
897 0615'          LD      HL,A
898 0616'          LD      HL,A
899 0617'          LD      HL,A
900 0618'          LD      HL,A
901 0619'          LD      HL,A
902 0620'          LD      HL,A
903 0621'          LD      HL,A
904 0622'          LD      HL,A
905 0623'          LD      HL,A
906 0624'          LD      HL,A
907 0625'          LD      HL,A
908 0626'          LD      HL,A
909 0627'          LD      HL,A
910 0628'          LD      HL,A
911 0629'          LD      HL,A
912 0630'          LD      HL,A
913 0631'          LD      HL,A
914 0632'          LD      HL,A
915 0633'          LD      HL,A
916 0634'          LD      HL,A
917 0635'          LD      HL,A
918 0636'          LD      HL,A
919 0637'          LD      HL,A
920 0638'          LD      HL,A
921 0639'          LD      HL,A
922 0640'          LD      HL,A
923 0641'          LD      HL,A
924 0642'          LD      HL,A
925 0643'          LD      HL,A
926 0644'          LD      HL,A
927 0645'          LD      HL,A
928 0646'          LD      HL,A
929 0647'          LD      HL,A
930 0648'          LD      HL,A
931 0649'          LD      HL,A
932 0650'          LD      HL,A
933 0651'          LD      HL,A
934 0652'          LD      HL,A
935 0653'          LD      HL,A
936 0654'          LD      HL,A
937 0655'          LD      HL,A
938 0656'          LD      HL,A
939 0657'          LD      HL,A
940 0658'          LD      HL,A
941 0659'          LD      HL,A
942 0660'          LD      HL,A
943 0661'          LD      HL,A
944 0662'          LD      HL,A
945 0663'          LD      HL,A
946 0664'          LD      HL,A
947 0665'          LD      HL,A
948 0666'          LD      HL,A
949 0667'          LD      HL,A
950 0668'          LD      HL,A
951 0669'          LD      HL,A
952 0670'          LD      HL,A
953 0671'          LD      HL,A
954 0672'          LD      HL,A
955 0673'          LD      HL,A
956 0674'          LD      HL,A
957 0675'          LD      HL,A
958 0676'          LD      HL,A
959 0677'          LD      HL,A
960 0678'          LD      HL,A
961 0679'          LD      HL,A
962 0680'          LD      HL,A
963 0681'          LD      HL,A
964 0682'          LD      HL,A
965 0683'          LD      HL,A
966 0684'          LD      HL,A
967 0685'          LD      HL,A
968 0686'          LD      HL,A
969 0687'          LD      HL,A
970 0688'          LD      HL,A
971 0689'          LD      HL,A
972 0690'          LD      HL,A
973 0691'          LD      HL,A
974 0692'          LD      HL,A
975 0693'          LD      HL,A
976 0694'          LD      HL,A
977 0695'          LD      HL,A
978 0696'          LD      HL,A
979 0697'          LD      HL,A
980 0698'          LD      HL,A
981 0699'          LD      HL,A
982 0700'          LD      HL,A
983 0701'          LD      HL,A
984 0702'          LD      HL,A
985 0703'          LD      HL,A
986 0704'          LD      HL,A
987 0705'          LD      HL,A
988 0706'          LD      HL,A
989 0707'          LD      HL,A
990 0708'          LD      HL,A
991 0709'          LD      HL,A
992 0710'          LD      HL,A
993 0711'          LD      HL,A
994 0712'          LD      HL,A
995 0713'          LD      HL,A
996 0714'          LD      HL,A
997 0715'          LD      HL,A
998 0716'          LD      HL,A
999 0717'          LD      HL,A

```

```

1222 0783' C0          RET  NZ
1223 0784' 2A ***     LD  HL,(PTRC0)
1224 0785' 3A ***     LD  A,(SEDM0)
1225 0786' FE 02      CP  2
1226 0787' C0          RET  Z
1227 0788' 2A ***     LD  HL,(PTR0T)
1228 0789' FE 03      CP  3
1229 0790' C0          RET  Z
1230 0791' 2A ***     LD  HL,(PTRW0)
1231 0792' C0          RET
1232 0793'
1233 0794'
1234 0795'
1235 0796'
1236 0797' 75         INCPC; PUSH  AF
1237 0798' E5         PUSH HL
1238 0799' 3A ***     LD  A,(SEDM0)
1239 079A' FE 03      CP  3
1240 079B' 28 00      JR  2,INCPC1
1241 079C' FE 04      CP  4
1242 079D' 28 12      JR  2,INCPC2
1243 079E'
1244 079F' 2A ***     LD  HL,(PTRC0);CSFG
1245 07A0' 23 ***     INC HL
1246 07A1' 22 ***     LD  (PTRC0),HL
1247 07A2' 10 10      JR  INCPC3
1248 07A3'
1249 07A4' 2A ***     INCPC1: LD  HL,(PTR0T);DSEG
1250 07A5' 23 ***     INC HL
1251 07A6' 22 ***     LD  (PTR0T),HL
1252 07A7' 10 07      JR  INCPC3
1253 07A8'
1254 07A9' 2A ***     INCPC2: LD  HL,(PTRW0);WSEG
1255 07AA' 23 ***     INC HL
1256 07AB' 22 ***     LD  (PTRW0),HL
1257 07AC'
1258 07AD' 2A ***     INCPC3: LD  HL,(PTRFC)
1259 07AE' 23 ***     INC HL
1260 07AF' 32 ***     LD  (PTRFC),HL
1261 07B0' 7C 01      POP  HL
1262 07B1' 7C 01      POP  AF
1263 07B2' C0          RET
1264 07B3'
1265 07B4'
1266 07B5'
1267 07B6' 13 ***     ERK0: LD  DE,MSG0
1268 07B7' C0 25 1F    CALL  MSG1
1269 07B8' 1A ***     LD  A,(C1)
1270 07B9' C0 21 1F    CALL  PTRNA
1271 07BA' C0 2E 1F    CALL  LITNL
1272 07BB' C0          RET
1273 07BC'
1274 07BD'
1275 07BE'
1276 07BF' 11 ***     ERK1: PUSH HL
1277 07C0' C0 25 1F    LD  DE,MSG1
1278 07C1' C0 21 1F    CALL  MSG1
1279 07C2' C0 2E 1F    CALL  PTRNL
1280 07C3' C0 2E 1F    CALL  LITNL
1281 07C4' C0          RET
1282 07C5'
1283 07C6'
1284 07C7'
1285 07C8'
1286 07C9'
1287 07CA' 13 ***     ERK2: LD  DE,MSG2
1288 07CB' C0 25 1F    CALL  MSG2
1289 07CC' C0 21 1F    CALL  MSG2
1290 07CD' C0 2E 1F    CALL  PTRNL
1291 07CE' C0 2E 1F    CALL  LITNL
1292 07CF' C0          RET
1293 07D0'
1294 07D1'
1295 07D2'
1296 07D3'
1297 07D4'
1298 07D5'
1299 07D6'
1300 07D7'
1301 07D8'
1302 07D9'
1303 07DA'
1304 07DB'
1305 07DC'
1306 07DD'
1307 07DE'
1308 07DF'
1309 07E0'
1310 07E1'
1311 07E2'
1312 07E3'
1313 07E4'
1314 07E5'
1315 07E6'
1316 07E7'
1317 07E8'
1318 07E9'
1319 07EA'
1320 07EB'
1321 07EC'
1322 07ED'
1323 07EE'
1324 07EF'
1325 07F0'
1326 07F1'
1327 07F2'
1328 07F3'
1329 07F4'
1330 07F5'
1331 07F6'
1332 07F7'
1333 07F8'
1334 07F9'
1335 07FA'
1336 07FB'
1337 07FC'
1338 07FD'
1339 07FE'
1340 07FF'
1341 0800'
1342 0801'
1343 0802'
1344 0803'
1345 0804'
1346 0805'
1347 0806'
1348 0807'
1349 0808'
1350 0809'
1351 080A'
1352 080B'
1353 080C'
1354 080D'
1355 080E'
1356 080F'
1357 0810'
1358 0811'
1359 0812'
1360 0813'
1361 0814'
1362 0815'
1363 0816'
1364 0817'
1365 0818'
1366 0819'
1367 081A'
1368 081B'
1369 081C'
1370 081D'
1371 081E'
1372 081F'
1373 0820'
1374 0821'
1375 0822'
1376 0823'
1377 0824'
1378 0825'
1379 0826'
1380 0827'
1381 0828'
1382 0829'
1383 082A'
1384 082B'
1385 082C'
1386 082D'
1387 082E'
1388 082F'
1389 0830'
1390 0831'
1391 0832'
1392 0833'
1393 0834'
1394 0835'
1395 0836'
1396 0837'
1397 0838'
1398 0839'
1399 083A'
1400 083B'
1401 083C'
1402 083D'
1403 083E'
1404 083F'
1405 0840'
1406 0841'
1407 0842'
1408 0843'
1409 0844'
1410 0845'
1411 0846'
1412 0847'
1413 0848'
1414 0849'
1415 084A'
1416 084B'
1417 084C'
1418 084D'
1419 084E'
1420 084F'
1421 0850'
1422 0851'
1423 0852'
1424 0853'
1425 0854'
1426 0855'
1427 0856'
1428 0857'
1429 0858'
1430 0859'
1431 085A'
1432 085B'
1433 085C'
1434 085D'
1435 085E'
1436 085F'
1437 0860'
1438 0861'
1439 0862'
1440 0863'
1441 0864'
1442 0865'
1443 0866'
1444 0867'
1445 0868'
1446 0869'
1447 086A'
1448 086B'
1449 086C'
1450 086D'
1451 086E'
1452 086F'
1453 0870'
1454 0871'
1455 0872'
1456 0873'
1457 0874'
1458 0875'
1459 0876'
1460 0877'
1461 0878'
1462 0879'
1463 087A'
1464 087B'
1465 087C'
1466 087D'
1467 087E'
1468 087F'
1469 0880'
1470 0881'
1471 0882'
1472 0883'
1473 0884'
1474 0885'
1475 0886'
1476 0887'
1477 0888'
1478 0889'
1479 088A'
1480 088B'
1481 088C'
1482 088D'
1483 088E'
1484 088F'
1485 0890'
1486 0891'
1487 0892'
1488 0893'
1489 0894'
1490 0895'
1491 0896'
1492 0897'
1493 0898'
1494 0899'
1495 089A'
1496 089B'
1497 089C'
1498 089D'
1499 089E'
1500 089F'
1501 0890'
1502 0891'
1503 0892'
1504 0893'
1505 0894'
1506 0895'
1507 0896'
1508 0897'
1509 0898'
1510 0899'
1511 089A'
1512 089B'
1513 089C'
1514 089D'
1515 089E'
1516 089F'
1517 0890'
1518 0891'
1519 0892'
1520 0893'
1521 0894'
1522 0895'
1523 0896'
1524 0897'
1525 0898'
1526 0899'
1527 089A'
1528 089B'
1529 089C'
1530 089D'
1531 089E'
1532 089F'
1533 0890'
1534 0891'
1535 0892'
1536 0893'
1537 0894'
1538 0895'
1539 0896'
1540 0897'
1541 0898'
1542 0899'
1543 089A'
1544 089B'
1545 089C'
1546 089D'
1547 089E'
1548 089F'
1549 0890'
1550 0891'
1551 0892'
1552 0893'
1553 0894'
1554 0895'
1555 0896'
1556 0897'
1557 0898'
1558 0899'
1559 089A'
1560 089B'
1561 089C'
1562 089D'
1563 089E'
1564 089F'
1565 0890'
1566 0891'
1567 0892'
1568 0893'
1569 0894'
1570 0895'
1571 0896'
1572 0897'
1573 0898'
1574 0899'
1575 089A'
1576 089B'
1577 089C'
1578 089D'
1579 089E'
1580 089F'
1581 0890'
1582 0891'
1583 0892'
1584 0893'
1585 0894'
1586 0895'
1587 0896'
1588 0897'
1589 0898'
1590 0899'
1591 089A'
1592 089B'
1593 089C'
1594 089D'
1595 089E'
1596 089F'
1597 0890'
1598 0891'
1599 0892'
1600 0893'
1601 0894'
1602 0895'
1603 0896'
1604 0897'
1605 0898'
1606 0899'
1607 089A'
1608 089B'
1609 089C'
1610 089D'
1611 089E'
1612 089F'
1613 0890'
1614 0891'
1615 0892'
1616 0893'
1617 0894'
1618 0895'
1619 0896'
1620 0897'
1621 0898'
1622 0899'
1623 089A'
1624 089B'
1625 089C'
1626 089D'
1627 089E'
1628 089F'
1629 0890'
1630 0891'
1631 0892'
1632 0893'
1633 0894'
1634 0895'
1635 0896'
1636 0897'
1637 0898'
1638 0899'
1639 089A'
1640 089B'
1641 089C'
1642 089D'
1643 089E'
1644 089F'
1645 0890'
1646 0891'
1647 0892'
1648 0893'
1649 0894'
1650 0895'
1651 0896'
1652 0897'
1653 0898'
1654 0899'
1655 089A'
1656 089B'
1657 089C'
1658 089D'
1659 089E'
1660 089F'
1661 0890'
1662 0891'
1663 0892'
1664 0893'
1665 0894'
1666 0895'
1667 0896'
1668 0897'
1669 0898'
1670 0899'
1671 089A'
1672 089B'
1673 089C'
1674 089D'
1675 089E'
1676 089F'
1677 0890'
1678 0891'
1679 0892'
1680 0893'
1681 0894'
1682 0895'
1683 0896'
1684 0897'
1685 0898'
1686 0899'
1687 089A'
1688 089B'
1689 089C'
1690 089D'
1691 089E'
1692 089F'
1693 0890'
1694 0891'
1695 0892'
1696 0893'
1697 0894'
1698 0895'
1699 0896'
1700 0897'
1701 0898'
1702 0899'
1703 089A'
1704 089B'
1705 089C'
1706 089D'
1707 089E'
1708 089F'
1709 0890'
1710 0891'
1711 0892'
1712 0893'
1713 0894'
1714 0895'
1715 0896'
1716 0897'
1717 0898'
1718 0899'
1719 089A'
1720 089B'
1721 089C'
1722 089D'
1723 089E'
1724 089F'
1725 0890'
1726 0891'
1727 0892'
1728 0893'
1729 0894'
1730 0895'
1731 0896'
1732 0897'
1733 0898'
1734 0899'
1735 089A'
1736 089B'
1737 089C'
1738 089D'
1739 089E'
1740 089F'
1741 0890'
1742 0891'
1743 0892'
1744 0893'
1745 0894'
1746 0895'
1747 0896'
1748 0897'
1749 0898'
1750 0899'
1751 089A'
1752 089B'
1753 089C'
1754 089D'
1755 089E'
1756 089F'
1757 0890'
1758 0891'
1759 0892'
1760 0893'
1761 0894'
1762 0895'
1763 0896'
1764 0897'
1765 0898'
1766 0899'
1767 089A'
1768 089B'
1769 089C'
1770 089D'
1771 089E'
1772 089F'
1773 0890'
1774 0891'
1775 0892'
1776 0893'
1777 0894'
1778 0895'
1779 0896'
1780 0897'
1781 0898'
1782 0899'
1783 089A'
1784 089B'
1785 089C'
1786 089D'
1787 089E'
1788 089F'
1789 0890'
1790 0891'
1791 0892'
1792 0893'
1793 0894'
1794 0895'
1795 0896'
1796 0897'
1797 0898'
1798 0899'
1799 089A'
1800 089B'
1801 089C'
1802 089D'
1803 089E'
1804 089F'
1805 0890'
1806 0891'
1807 0892'
1808 0893'
1809 0894'
1810 0895'
1811 0896'
1812 0897'
1813 0898'
1814 0899'
1815 089A'
1816 089B'
1817 089C'
1818 089D'
1819 089E'
1820 089F'
1821 0890'
1822 0891'
1823 0892'
1824 0893'
1825 0894'
1826 0895'
1827 0896'
1828 0897'
1829 0898'
1830 0899'
1831 089A'
1832 089B'
1833 089C'
1834 089D'
1835 089E'
1836 089F'
1837 0890'
1838 0891'
1839 0892'
1840 0893'
1841 0894'
1842 0895'
1843 0896'
1844 0897'
1845 0898'
1846 0899'
1847 089A'
1848 089B'
1849 089C'
1850 089D'
1851 089E'
1852 089F'
1853 0890'
1854 0891'
1855 0892'
1856 0893'
1857 0894'
1858 0895'
1859 0896'
1860 0897'
1861 0898'
1862 0899'
1863 089A'
1864 089B'
1865 089C'
1866 089D'
1867 089E'
1868 089F'
1869 0890'
1870 0891'
1871 0892'
1872 0893'
1873 0894'
1874 0895'
1875 0896'
1876 0897'
1877 0898'
1878 0899'
1879 089A'
1880 089B'
1881 089C'
1882 089D'
1883 089E'
1884 089F'
1885 0890'
1886 0891'
1887 0892'
1888 0893'
1889 0894'
1890 0895'
1891 0896'
1892 0897'
1893 0898'
1894 0899'
1895 089A'
1896 089B'
1897 089C'
1898 089D'
1899 089E'
1900 089F'
1901 0890'
1902 0891'
1903 0892'
1904 0893'
1905 0894'
1906 0895'
1907 0896'
1908 0897'
1909 0898'
1910 0899'
1911 089A'
1912 089B'
1913 089C'
1914 089D'
1915 089E'
1916 089F'
1917 0890'
1918 0891'
1919 0892'
1920 0893'
1921 0894'
1922 0895'
1923 0896'
1924 0897'
1925 0898'
1926 0899'
1927 089A'
1928 089B'
1929 089C'
1930 089D'
1931 089E'
1932 089F'
1933 0890'
1934 0891'
1935 0892'
1936 0893'
1937 0894'
1938 0895'
1939 0896'
1940 0897'
1941 0898'
1942 0899'
1943 089A'
1944 089B'
1945 089C'
1946 089D'
1947 089E'
1948 089F'
1949 0890'
1950 0891'
1951 0892'
1952 0893'
1953 0894'
1954 0895'
1955 0896'
1956 0897'
1957 0898'
1958 0899'
1959 089A'
1960 089B'
1961 089C'
1962 089D'
1963 089E'
1964 089F'
1965 0890'
1966 0891'
1967 0892'
1968 0893'
1969 0894'
1970 0895'
1971 0896'
1972 0897'
1973 0898'
1974 0899'
1975 089A'
1976 089B'
1977 089C'
1978 089D'
1979 089E'
1980 089F'
1981 0890'
1982 0891'
1983 0892'
1984 0893'
1985 0894'
1986 0895'
1987 0896'
1988 0897'
1989 0898'
1990 0899'
1991 089A'
1992 089B'
1993 089C'
1994 089D'
1995 089E'
1996 089F'
1997 0890'
1998 0891'
1999 0892'
2000 0893'
2001 0894'
2002 0895'
2003 0896'
2004 0897'
2005 0898'
2006 0899'
2007 089A'
2008 089B'
2009 089C'
2010 089D'
2011 089E'
2012 089F'
2013 0890'
2014 0891'
2015 0892'
2016 0893'
2017 0894'
2018 0895'
2019 0896'
2020 0897'
2021 0898'
2022 0899'
2023 089A'
2024 089B'
2025 089C'
2026 089D'
2027 089E'
2028 089F'
2029 0890'
2030 0891'
2031 0892'
2032 0893'
2033 0894'
2034 0895'
2035 0896'
2036 0897'
2037 0898'
2038 0899'
2039 089A'
2040 089B'
2041 089C'
2042 089D'
2043 089E'
2044 089F'
2045 0890'
2046 0891'
2047 0892'
2048 0893'
2049 0894'
2050 0895'
2051 0896'
2052 0897'
2053 0898'
2054 0899'
2055 089A'
2056 089B'
2057 089C'
2058 089D'
2059 089E'
2060 089F'
2061 0890'
2062 0891'
2063 0892'
2064 0893'
2065 0894'
2066 0895'
2067 0896'
2068 0897'
2069 0898'
2070 0899'
2071 089A'
2072 089B'
2073 089C'
2074 089D'
2075 089E'
2076 089F'
2077 0890'
2078 0891'
2079 0892'
2080 0893'
2081 0894'
2082 0895'
2083 0896'
2084 0897'
2085 0898'
2086 0899'
2087 089A'
2088 089B'
2089 089C'
2090 089D'
2091 089E'
2092 089F'
2093 0890'
2094 0891'
2095 0892'
2096 0893'
2097 0894'
2098 0895'
2099 0896'
2100 0897'
2101 0898'
2102 0899'
2103 089A'
2104 089B'
2105 089C'
2106 089D'
2107 089E'
2108 089F'
2109 0890'
2110 0891'
2111 0892'
2112 0893'
2113 0894'
2114 0895'
2115 0896'
2116 0897'
2117 0898'
2118 0899'
2119 089A'
2120 089B'
2121 089C'
2122 089D'
2123 089E'
2124 089F'
2125 0890'
2126 0891'
2127 0892'
2128 0893'
2129 0894'
2130 0895'
2131 0896'
2132 0897'
2133 0898'
2134 0899'
2135 089A'
2136 089B'
2137 089C'
2138 089D'
2139 089E'
2140 089F'
2141 0890'
2142 0891'
2143 0892'
2144 0893'
2145 0894'
2146 0895'
2147 0896'
2148 0897'
2149 0898'
2150 0899'
2151 089A'
2152 089B'
2153 089C'
2154 089D'
2155 089E'
2156 089F'
2157 0890'
2158 0891'
2159 0892'
2160 0893'
2161 0894'
2162 0895'
2163 0896'
2164 0897'
2165 0898'
2166 0899'
2167 089A'
2168 089B'
2169 089C'
2170 089D'
2171 089E'
2172 089F'
2173 0890'
2174 0891'
2175 0892'
2176 0893'
2177 0894'
2178 0895'
2179 0896'
2180 0897'
2181 0898'
2182 0899'
2183 089A'
2184 089B'
2185 089C'
2186 089D'
2187 089E'
2188 089F'
2189 0890'
2190 0891'
2191 0892'
2192 0893'
2193 0894'
2194 0895'
2195 0896'
2196 0897'
2197 0898'
2198 0899'
2199 089A'
2200 089B'
2201 089C'
2202 089D'
2203 089E'
2204 089F'
2205 0890'
2206 0891'
2207 0892'
2208 0893'
2209 0894'
2210 0895'
2211 0896'
2212 0897'
2213 0898'
2214 0899'
2215 089A'
2216 089B'
2217 089C'
2218 089D'
2219 089E'
2220 089F'
2221 0890'
2222 0891'
2223 0892'
2224 0893'
2225 0894'
2226 0895'
2227 0896'
2228 0897'
2229 0898'
2230 0899'
2231 089A'
2232 089B'
2233 089C'
2234 089D'
2235 089E'
2236 089F'
2237 0890'
2238 0891'
2239 0892'
2240 0893'
2241 0894'
2242 0895'
2243 0896'
2244 0897'
2245 0898'
2246 0899'
2247 089A'
2248 089B'
2249 089C'
2250 089D'
2251 089E'
2252 089F'
2253 0890'
2254 0891'
2255 0892'
2256 0893'
2257 0894'
2258 0895'
2259 0896'
2260 0897'
2261 0898'
2262 0899'
2263 089A'
2264 089B'
2265 089C'
2266 089D'
2267 089E'
2268 089F'
2269 0890'
2270 0891'
2271 0892'
2272 0893'
2273 0894'
2274 0895'
2275 0896'
2276 0897'
2277 0898'
2278 0899'
2279 089A'
2280 089B'
2281 089C'
2282 089D'
2283 089E'
2284 089F'
2285 0890'
2286 0891'
2287 0892'
2288 0893'
2289 0894'
2290 0895'
2291 0896'
2292 0897'
2293 0898'
2294 0899'
2295 089A'
2
```



```

611 833F C9 3C      SRL      H      ;HL/2
612 8341 C9 1D      RR      L      ;HL/4
613 8343 C9 3C      RR      L      ;HL/4
614 8345 C9 1D      SRL      H      ;HL/8
615 8347 C9 3C      SRL      H      ;HL/8
616 8349 C9 1D      RR      L      ;HL/8
617 834B C9 3C      SRL      H      ;HL/8
618 834D C9 1D      RR      L      ;HL/8
619 834F C9 1D      POP      HL
620 8350 E1         RET
621 8351 C9         RET
622 8352          ;
623 8352          ; Works
624 8352          ;
625 8352          ;
626 8352          ;
627 8352          ;
628 8352          ;
629 8352          ;
630 8352          ;
631 8352          ;
632 8352          ;
633 8352          ;
634 8352          ;
635 8352          ;
636 8352          ;
637 8352          ;
638 8352          ;
639 8352          ;
640 8352          ;
641 8352          ;
642 8352          ;
643 8352          ;
644 8352          ;
645 8352          ;
646 8352          ;
647 8352          ;
648 8352          ;
649 8352          ;
650 8352          ;
651 8352          ;
652 8352          ;
653 8352          ;
654 8352          ;
655 8352          ;
656 8352          ;
657 8352          ;
658 8352          ;
659 8352          ;
660 8352          ;
661 8352          ;
662 8352          ;
663 8352          ;
664 8352          ;
665 8352          ;
666 8352          ;
667 8352          ;
668 8352          ;
669 8352          ;
670 8352          ;
671 8352          ;
672 8352          ;
673 8352          ;
674 8352          ;
675 8352          ;
676 8352          ;
677 8352          ;
678 8352          ;
679 8352          ;
680 8352          ;
681 8352          ;
682 8352          ;
683 8352          ;
684 8352          ;
685 8352          ;
686 8352          ;
687 8352          ;
688 8352          ;
689 8352          ;
690 8352          ;
691 8352          ;
692 8352          ;
693 8352          ;
694 8352          ;
695 8352          ;
696 8352          ;
697 8352          ;
698 8352          ;
699 8352          ;
700 8352          ;
701 8352          ;
702 8352          ;
703 8352          ;
704 8352          ;
705 8352          ;
706 8352          ;
707 8352          ;
708 8352          ;
709 8352          ;
710 8352          ;
711 8352          ;
712 8352          ;
713 8352          ;
714 8352          ;
715 8352          ;
716 8352          ;
717 8352          ;
718 8352          ;
719 8352          ;
720 8352          ;
721 8352          ;
722 8352          ;
723 8352          ;
724 8352          ;
725 8352          ;
726 8352          ;
727 8352          ;
728 8352          ;
729 8352          ;
730 8352          ;
731 8352          ;
732 8352          ;
733 8352          ;
734 8352          ;
735 8352          ;
736 8352          ;
737 8352          ;
738 8352          ;
739 8352          ;
740 8352          ;
741 8352          ;
742 8352          ;
743 8352          ;
744 8352          ;
745 8352          ;
746 8352          ;
747 8352          ;
748 8352          ;
749 8352          ;
750 8352          ;
751 8352          ;
752 8352          ;
753 8352          ;
754 8352          ;
755 8352          ;
756 8352          ;
757 8352          ;
758 8352          ;
759 8352          ;
760 8352          ;
761 8352          ;
762 8352          ;
763 8352          ;
764 8352          ;
765 8352          ;
766 8352          ;
767 8352          ;
768 8352          ;
769 8352          ;
770 8352          ;
771 8352          ;
772 8352          ;
773 8352          ;
774 8352          ;
775 8352          ;
776 8352          ;
777 8352          ;
778 8352          ;
779 8352          ;
780 8352          ;
781 8352          ;
782 8352          ;
783 8352          ;
784 8352          ;
785 8352          ;
786 8352          ;
787 8352          ;
788 8352          ;
789 8352          ;
790 8352          ;
791 8352          ;
792 8352          ;
793 8352          ;
794 8352          ;
795 8352          ;
796 8352          ;
797 8352          ;
798 8352          ;
799 8352          ;
800 8352          ;
801 8352          ;
802 8352          ;
803 8352          ;
804 8352          ;
805 8352          ;
806 8352          ;
807 8352          ;
808 8352          ;
809 8352          ;
810 8352          ;
811 8352          ;
812 8352          ;
813 8352          ;
814 8352          ;
815 8352          ;
816 8352          ;
817 8352          ;
818 8352          ;
819 8352          ;
820 8352          ;
821 8352          ;
822 8352          ;
823 8352          ;
824 8352          ;
825 8352          ;
826 8352          ;
827 8352          ;
828 8352          ;
829 8352          ;
830 8352          ;
831 8352          ;
832 8352          ;
833 8352          ;
834 8352          ;
835 8352          ;
836 8352          ;
837 8352          ;
838 8352          ;
839 8352          ;
840 8352          ;
841 8352          ;
842 8352          ;
843 8352          ;
844 8352          ;
845 8352          ;
846 8352          ;
847 8352          ;
848 8352          ;
849 8352          ;
850 8352          ;
851 8352          ;
852 8352          ;
853 8352          ;
854 8352          ;
855 8352          ;
856 8352          ;
857 8352          ;
858 8352          ;
859 8352          ;
860 8352          ;
861 8352          ;
862 8352          ;
863 8352          ;
864 8352          ;
865 8352          ;
866 8352          ;
867 8352          ;
868 8352          ;
869 8352          ;
870 8352          ;
871 8352          ;
872 8352          ;
873 8352          ;
874 8352          ;
875 8352          ;
876 8352          ;
877 8352          ;
878 8352          ;
879 8352          ;
880 8352          ;
881 8352          ;
882 8352          ;
883 8352          ;
884 8352          ;
885 8352          ;
886 8352          ;
887 8352          ;
888 8352          ;
889 8352          ;
890 8352          ;
891 8352          ;
892 8352          ;
893 8352          ;
894 8352          ;
895 8352          ;
896 8352          ;
897 8352          ;
898 8352          ;
899 8352          ;
900 8352          ;
901 8352          ;
902 8352          ;
903 8352          ;
904 8352          ;
905 8352          ;
906 8352          ;
907 8352          ;
908 8352          ;
909 8352          ;
910 8352          ;
911 8352          ;
912 8352          ;
913 8352          ;
914 8352          ;
915 8352          ;
916 8352          ;
917 8352          ;
918 8352          ;
919 8352          ;
920 8352          ;
921 8352          ;
922 8352          ;
923 8352          ;
924 8352          ;
925 8352          ;
926 8352          ;
927 8352          ;
928 8352          ;
929 8352          ;
930 8352          ;
931 8352          ;
932 8352          ;
933 8352          ;
934 8352          ;
935 8352          ;
936 8352          ;
937 8352          ;
938 8352          ;
939 8352          ;
940 8352          ;
941 8352          ;
942 8352          ;
943 8352          ;
944 8352          ;
945 8352          ;
946 8352          ;
947 8352          ;
948 8352          ;
949 8352          ;
950 8352          ;
951 8352          ;
952 8352          ;
953 8352          ;
954 8352          ;
955 8352          ;
956 8352          ;
957 8352          ;
958 8352          ;
959 8352          ;
960 8352          ;
961 8352          ;
962 8352          ;
963 8352          ;
964 8352          ;
965 8352          ;
966 8352          ;
967 8352          ;
968 8352          ;
969 8352          ;
970 8352          ;
971 8352          ;
972 8352          ;
973 8352          ;
974 8352          ;
975 8352          ;
976 8352          ;
977 8352          ;
978 8352          ;
979 8352          ;
980 8352          ;
981 8352          ;
982 8352          ;
983 8352          ;
984 8352          ;
985 8352          ;
986 8352          ;
987 8352          ;
988 8352          ;
989 8352          ;
990 8352          ;
991 8352          ;
992 8352          ;
993 8352          ;
994 8352          ;
995 8352          ;
996 8352          ;
997 8352          ;
998 8352          ;
999 8352          ;
1000 8352          ;

```

```

830 0000"          ;Start Address.
831 0001"          ;Exec Address.
832 0002"          ;First Cluster.
833 0003"          ;Cluster Table.
834 0004"          ;The Login disk.
835 0005"          ;The FILE Pointer.
836 0006"          ;Record No. Which Have The BT
837 0007"          ;Address Where On IFORM1110
838 0008"          ;The Number of Records The File
839 0009"          ;
840 0010"          ;
841 0011"          ;
842 0012"          ;
843 0013"          ;
844 0014"          ;
845 0015"          ;
846 0016"          ;
847 0017"          ;
848 0018"          ;
849 0019"          ;
850 0020"          ;
851 0021"          ;
852 0022"          ;
853 0023"          ;
854 0024"          ;
855 0025"          ;
856 0026"          ;
857 0027"          ;
858 0028"          ;
859 0029"          ;
860 0030"          ;
861 0031"          ;
862 0032"          ;
863 0033"          ;
864 0034"          ;
865 0035"          ;
866 0036"          ;
867 0037"          ;
868 0038"          ;
869 0039"          ;
870 0040"          ;
871 0041"          ;
872 0042"          ;
873 0043"          ;
874 0044"          ;
875 0045"          ;
876 0046"          ;
877 0047"          ;
878 0048"          ;
879 0049"          ;
880 0050"          ;
881 0051"          ;
882 0052"          ;
883 0053"          ;
884 0054"          ;
885 0055"          ;
886 0056"          ;
887 0057"          ;
888 0058"          ;
889 0059"          ;
890 0060"          ;
891 0061"          ;
892 0062"          ;
893 0063"          ;
894 0064"          ;
895 0065"          ;
896 0066"          ;
897 0067"          ;
898 0068"          ;
899 0069"          ;
900 0070"          ;
901 0071"          ;
902 0072"          ;
903 0073"          ;
904 0074"          ;
905 0075"          ;
906 0076"          ;
907 0077"          ;
908 0078"          ;
909 0079"          ;
910 0080"          ;
911 0081"          ;
912 0082"          ;
913 0083"          ;
914 0084"          ;
915 0085"          ;
916 0086"          ;
917 0087"          ;
918 0088"          ;
919 0089"          ;
920 0090"          ;
921 0091"          ;
922 0092"          ;
923 0093"          ;
924 0094"          ;
925 0095"          ;
926 0096"          ;
927 0097"          ;
928 0098"          ;
929 0099"          ;
930 0100"          ;
931 0101"          ;
932 0102"          ;
933 0103"          ;
934 0104"          ;
935 0105"          ;
936 0106"          ;
937 0107"          ;
938 0108"          ;
939 0109"          ;
940 0110"          ;
941 0111"          ;
942 0112"          ;
943 0113"          ;
944 0114"          ;
945 0115"          ;
946 0116"          ;
947 0117"          ;
948 0118"          ;
949 0119"          ;
950 0120"          ;
951 0121"          ;
952 0122"          ;
953 0123"          ;
954 0124"          ;
955 0125"          ;
956 0126"          ;
957 0127"          ;
958 0128"          ;
959 0129"          ;
960 0130"          ;
961 0131"          ;
962 0132"          ;
963 0133"          ;
964 0134"          ;
965 0135"          ;
966 0136"          ;
967 0137"          ;
968 0138"          ;
969 0139"          ;
970 0140"          ;
971 0141"          ;
972 0142"          ;
973 0143"          ;
974 0144"          ;
975 0145"          ;
976 0146"          ;
977 0147"          ;
978 0148"          ;
979 0149"          ;
980 0150"          ;
981 0151"          ;
982 0152"          ;
983 0153"          ;
984 0154"          ;
985 0155"          ;
986 0156"          ;
987 0157"          ;
988 0158"          ;
989 0159"          ;
990 0160"          ;
991 0161"          ;
992 0162"          ;
993 0163"          ;
994 0164"          ;
995 0165"          ;
996 0166"          ;
997 0167"          ;
998 0168"          ;
999 0169"          ;
1000 0170"          ;

```

```

6 EQU 5000H
7 EQU 5F00H
8 EQU 5F00H
9 EQU 5F00H
10 EQU 5F00H
11 EQU 5F00H
12 EQU 5F00H
13 EQU 5F00H
14 EQU 5F00H
15 EQU 5F00H

```

リスト6

```

1 Header File For WZD
2 CSEG 3000H-
3 DSEG 6000H-
4
5
6
7
8
9

```

リスト5

```

1 Header File For WLK
2 CSEG 3000H-
3 DSEG 6000H-
4
5
6
7
8
9

```

全機種共通システムインデックス

■85年6月号

序論 共通化の試み

第1部 S-OS"MACE"

第2部 Lisp-85インタプリタ

第3部 チェックサムプログラム

■85年7月号

第4部 マシン語プログラム開発入門

第5部 エディタアセンブラZEDA

第6部 デバッグツールZaid

■85年8月号

第7部 ゲーム開発パッケージBEMS

第8部 ソースジェネレータZING

■85年9月号

インタラプト S-OS番外地

第9部 マシン語入力ツールMACINTO-S

第10部 Lisp-85入門(1)

■85年10月号

第11部 仮想マシンCAP-X85

連載 Lisp-85入門(2)

■85年11月号

連載 Lisp-85入門(3)

■85年12月号

第12部 Prolog-85発表

■86年1月号

第13部 リロケータブルのお話

第14部 FM音源サウンドエディタ

■86年2月号

第15部 S-OS"SWORD"

第16部 Prolog-85入門(1)

■86年3月号

第17部 magiFORTH発表

連載 Prolog-85入門(2)

■86年4月号

第18部 思考ゲームJEWEL

第19部 LIFE GAME

連載 基礎からのmagiFORTH

連載 Prolog-85入門(3)

■86年5月号

第20部 スクリンエディタE-MATE

連載 実戦演習magiFORTH

■86年6月号

第21部 Z80TRACER

第22部 magiFORTH TRACER

第23部 ディスクダンプ&エディタ

第24部 "SWORD" 2000 QD

連載 対話で学ぶ magiFORTH

特別付録 PC-8801版S-OS"SWORD"

■86年7月号

第25部 FM音源ミュージックシステム

付録 FM音源ボードの製作

連載 計算機アツプのmagiFORTH

特別付録 SMC-777版S-OS"SWORD"

■86年8月号

第26部 対局五目並べ

第27部 MZ-2500版S-OS"SWORD"

■86年9月号

第28部 FuzzyBASIC 発表

連載 明日に向かって magiFORTH

■86年10月号

第29部 ちょっと便利な拡張プログラム

第30部 ディスクモニタ DREAM

第31部 FuzzyBASIC 料理法(1)

■86年11月号

第32部 バズルゲーム HOTTAN

第33部 MAZE in MAZE

連載 FuzzyBASIC 料理法(2)

■86年12月号

第34部 CASL & COMET

連載 FuzzyBASIC 料理法(3)

■87年1月号

第35部 マシン語入力ツールMACINTO-C

連載 FuzzyBASIC 料理法(4)

■87年2月号

第36部 アドベンチャーゲーム MARMALADE

第37部 テキアベ作成ツール CONTEX

■87年3月号

第38部 魔法使いはアニメがお好き

第39部 アニメーションツール MAGE

付録 "SWORD" 再掲載と MAGIC の標準化

■87年4月号

第40部 INVADER GAME

第41部 TANGERINE

■87年5月号

第42部 S-OS"SWORD" 変身セット

第43部 MZ-700用 "SWORD" を QD 対応に

■87年6月号

インタラプト コンパイル物語

第44部 FuzzyBASIC コンパイラ

第45部 エディタアセンブラ ZEDA-3

■87年7月号

第46部 STORY MASTER

■87年8月号

第47部 バズルゲーム 碁石拾い

第48部 漢字出力パッケージ JACKWRITE

特別付録 FM-7/77版 S-OS"SWORD"

■87年9月号

第49部 リロケータブル逆アセンブラ Inside-R

特別付録 PC-8001/8801 版 S-OS"SWORD"

■87年10月号

第50部 tiny CORE WARS

第51部 FuzzyBASIC コンパイラの拡張

第52部 Xturbo 版 S-OS"SWORD"

■87年11月号

序論 神話のなかのマイクロコンピュータ

付録 S-OS の仲間たち

第53部 もうひとつの FuzzyBASIC 入門

第54部 ファイルアロケータ&ローダ

インタ

人工知能の冒険

完全な真空

毛色の変った本を出すとして有名な国書刊行会から出されている『完全な真空』という、おかしな本をぜひとも皆さんに紹介しようと思います。著者はスタニスワフ・レムでして、タルコフスキーの撮った「惑星ソラリス」という映画の原作者として有名です。レムはSF作家としてきわめて有名であり、「最高のSF作家」とさえ呼ばれているそうです。

この『完全な真空』は、本当はこの世に存在しない本を、まるで存在するかのように出版社や作者名まででっち上げたうえ、それらの本それぞれに対する書評をまた自分で書いているというものです。ひとことでいえば、架空書評集というところでしよう。

全部で16冊の架空の書物が取り上げられているのですが、おもしろいものとおもしろくないものの差がきわめて大きく感じられました。あまり興味がもてなかったのが、『親衛隊少将ルイ16世』や『白痴』のように、なにかスケールの異常に大きい大作の概略を示したようなものです。逆に、3度も4度も読んでしまったのが、「最高のSF作家」こそが書きうるというようなものです。

実在する1冊の本

正確にいうならば、この本に収録されている16冊の架空書物のうち、先頭に取り上げられている1冊だけは実在します。それは、この本『完全な真空』自体です。そこでは、まるで別の人が書いたように、「レム氏は……」などとしらばっくれて書いています。しかもさらに、その文章の中で、その文章そのものを引用することまで行い、一層混乱の度をわざと高めています。

書き上げてもない本を作り出し、それを今度は評論家の立場で好き勝手に批評し、そうしてできた本をまた同じ本の中で批評するとは、いってみればなんとも書きとしてぜいたくなことをやっているのだろう

と思わず感じてしまいます。

・このように書評集の中でその書評集自体を取り上げるというのは、「再帰呼び出し」(リカーシブコール)を思い起こさせます。この例に見られるように、再帰呼び出し的なことは単にプログラムの中の関数の呼び出し方だけに限定された話ではありません。ネーミングの中に見られるごく簡単な例を示しましょう。UNIXオペレーティングシステムの発展版にそのスペルを引っ繰り返したXINU(ジーニュ)というのがありますが、これは次の文章の頭文字をとったものだそうです。

“XINU Is Not Unix.”

研究室にあるUNIXマシンのひとつ(CPUはSPARC)の名前を、SPARCを引っ繰り返したCRAPSとしているのですが、その名前の由来も無理やりこのXINUのように説明するならば、

“CRAPS Runs A Processor Sparc.”
(CRAPSはSparcプロセッサを駆動する)とでもいえばよいのでしょうか。

存在しえない小説

『完全な真空』の中で取り上げられている架空小説のうちのひとつに「とどのつまりは何も無し」というものがあります。この小説についてここで紹介し、読者の皆さんにああこういう小説なのかとわかってもらうことは、きわめてむずかしいことと思われれます。第一、僕自身どう考えても、このような小説がどのように存在し得るか、あまり想像がつかないからです。

まあとにかく、この小説の内容を紹介することにしましょう(無駄とわかっていても)。この小説の内容はないのです。といっても、真っ白な紙が並んでいるのではなく、しっかりと文章が並んでいるのです(もちろん、「何も無い」と1000回書かれているわけでもありません)。しかし、何も語ってはいないのです。

冒頭の文は「列車は着かなかった」となっています。そして、「誰か」が現れなかったあと、語りは非人称のまま、時は春で

もなく夏でもなく、無重力空間における愛されない女に関する考察によって第1章は閉じられます。

その後、この本に関する記述は抽象度を増します。「虚無の穴が不気味に大きくなってゆく」「思考しないことの流れ」「テキストはわれわれの所有していたものを次々と奪い取っていく」……。作品の最後ではもうこれ以上作品が続くかという疑念が沸き起こってきます。

そして、ついには「存在しないこと」は否定として存在することさえやめてしまうのです。文章の意味が失われると残るのは構文のみです。しかしその文法装置さえまいには空中分解してしまい、文章の途中、単語の途中でついにこの小説は終わってしまうのです(とまあちょっとだけ書いてみましたが、やはり徒労に終わったのでしょね?)。

でも、実際には存在しえない小説を仮想することこそ、この本の真価といえるでしょう。しかもなぜこのような小説がこの世に存在するかという意味づけもしっかりとなされています。要するに、小説家が誠実さを究極にまで追求したときに必然的に生まれる小説は、まさにこのようなものであるということです。小説家はありもしないことを書かなくてはならないのですが、もしそのような行為に良心の呵責を感じるような小説家が万一存在したならば、彼の取るべき道は2つだけ、筆を折るか、あるいは「とどのつまりは何も無い」小説を書くかということなのです。

このような小説を書く小説家の誠実さについて論じながらレムは、「私はそのような意味での誠実さからはいちばん遠いのだ」と含み笑いしていることでしょう。小説家が誠実さを求めることは、レムの行っている「ありもしない小説をでっち上げる」行為とちょうど正反対であるからなのです。

ところで、この世に存在しない小説の書評をした本を取り上げて、それをまた書評している僕自身の誠実さはいったいどうな

っているのでしょうか？ まあ、この『完全な真空』という本が存在しないのならば、それこそ賞賛に値するほどの不誠実さともいえるでしょうが、僕はまだまだ……。

知能の相対化

既成のとらわれた概念に対する鋭い風刺の効いた疑問は、この本のいろいろなところに見られます。「誤謬としての文化」では、まず、「文化は生物が生き残る邪魔にもならなければ、助けにもならないものである」という考えを否定します（これはまあ普通の主張といえましょう）。ところが次に主張されるのはきわめて刺激的な考えです。「文化というものは、自ら作り出した宗教、慣習、法、禁止、命令を通じて作用することにより、不十分なものを理想に、マイナスをプラスに、欠点や欠陥のあるものを完璧なものに作り変えるのだ」というのです。

あるいは別の書評では、知能というものに関して、人間の知能の絶対性というものに強い懐疑を示します。そしてこれは、「完全な真空」以外の彼の書物にも見られる一貫した態度のようです。人工知能という言葉は、最近ではごく当たり前に使われる言葉になってきたのですが、その際、知能は人間の頭脳こそが唯一もっているものであるということは、当然のこと、暗黙の了解事項であるように僕には感じられます。「ソラリス」のテーマ自体がそうであったように、レムはいつも人間のもっているものが知能として絶対唯一であるということへの疑問を提示しています。それどころか、この本を読むと人間の知能などは偶然の産物なのだという声さえ、きわめて皮肉的かつ間接的ではありますが、聞こえてきます。

この本が書かれたのはなんと1971年です（日本語訳が出たのは1989年）。その後10年くらいたって、いわゆるサイバーパンクといわれる新しい潮流が生まれて、人間の脳の神経細胞のクローズアップ、たとえば、直接、神経細胞をメディアとしてコミュニ

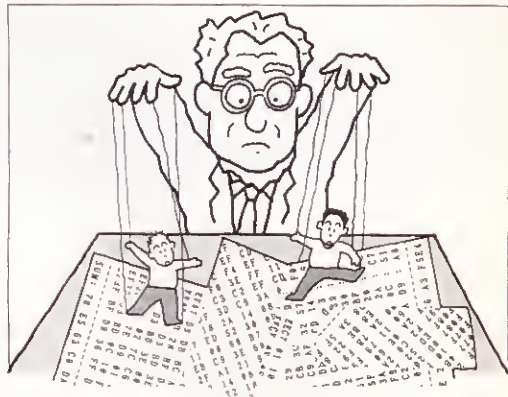
ケーションするという考えなどが生まれたわけです。次に紹介する架空書物評などを読むと、この本が今から20年も前に書かれたとは信じられない気がします。

「我は下僕ならずや」では、現実世界からはまったく切り離された神経細胞における電気パルスの伝達のみ構成される世界というものを、さらに独立させ、純粋化した世界を描いています。キーワードはパーソネティクス（理性ある生物の人工生産）なのだそうです。そのような世界を小説として描いているのではなく、実際にそのような世界を研究室の計算機内に作り上げたドブ教授がこの架空小説の著者なのです。

もうひとつ別の人工知能

「我は下僕ならずや」で記述されている世界といっても、完全に計算機の中の閉じた世界であり、まったく数学的に作り上げられたもののなのです。しかしそこには「住人」がいます。なぜそこに、知能をもつ生命体が住んでいるとみなせるかというところがミソであり、えんえんとページが費やされています。いわゆるシミュレーションのようにも思えるのですが、シミュレーションではありません、実体なのです。現実の物理的空間がないところになぜ知的生命体が想定できるのでしょうか？

このことについては、人間が住んでいるこの世というものが実は偶然の産物であり、数学的な世界の中にも、人間世界とまったく同じような現象が起こりうるということを執拗に述べています。偶然この世は3次元なのですが、彼ら「住人」の住む数学的世界ではそれらが任意に（ドブ教授によって）設定することができるということです。時間の進み具合も設定できます。ある種の具体化を遂げた数学は、完全に実体をもたぬほどに精神化した知性の生活空間となりえたのです。さらにレムは、この世や人間に特有なさざまな概念、たとえば、意



識、言語、進化などに関して、その脆弱性（もろくて弱いということ）を追求し、そして計算機の中に閉じ込められた世界でも同様の概念が存在するというのです。

この架空小説が最大に盛り上がるのは、「住人」たちの、創造主（つまりこの架空小説を書いているドブ教授）に関する議論です。何人かの「住人」たちが、いったい創造主はいるのかいないのか、いるのならば、今の我々とどういう関係にあるのかということをは話し合うのです。おもしろいのは、彼らの世界を述べているようで、いつの間にか、実は我々人間自身の問題と完全にオーバーラップしてくることです。

レムは実は計算機の中の人間が作り出した世界に生きる知的生命体を描きながら、実は、我々もまた上のレベルにある何者か（創造主）に操られているというような循環をも同時に描いているのでしょう。

知能機械といっても、人がもっているような知能だけを相手にしているのではもう古いのかもしれませんが、50年先、500年先をにらんで生きていく人は、ソラリスの海やサイバーパンクやパーソネティクスまでも包括したものとして、知能というものをイメージしていかなければならないのでしょう。

というわけで、本連載でも、総力を込めてというか、脱線しまくってというか、次回には、毛色の全然違う未知の領域に踏み込もうと思います。タイトルは、「超能力大実験：ここにも超能力者が！（仮称）」です。（こりゃとんだことになりそうだと感じつつ）来月をお楽しみに！

猫とコンピュータ サーチャーでござる

Takazawa Kyoko
高沢 恭子

ホンニャは体内に上等のセンサーがあるから、日に日に近づく灼熱の季節を、もう感じている。うすぐもりの空と湿った風にくるまれて、太陽はまだ休息しているのだ。

アイハラさんちのハチが、顔を天に向けて鼻をヒクヒクさせているけれど、あいつは犬だからまだ気づいてはいまい。そう彼は思う。猫の中にもにぶいのはいる。背中に座布団をのせたようなデザインのザブなんか、おデコのハエにも気がつかないほど感度が悪い。

でも、ホンニャにはわかるのだ。ひかえめなようすを見せながら、けっこう大きな群れをつくって咲いているアジサイの花のかげで、もう夏は光りはじめていることを……。

光る床

つゆ明けはまだ先のことなのに、気温の上昇につれて、ホンニャの体はアメがとけるようにだんだん伸びていく。彼の体の伸び縮みは温度計のようだ。そしてわが家の木の床板としいに仲良しになって、ダラリ、ペタリとはりついて過ごす時間がふえていく。

床張りをほどこしたものを、このごろではフローリング(flooring)としゃれた呼びかたをするらしいが、正方形をつなぎ合わせた木目の床は、夏の午後なら、猫でなくても寝そべってみたいくなる。

木の性質のふしぎさは、夏はひんやりとした感触でやすらぎを与えてくれるのに、冬は冬で独特のあたたかさをただよわせることだ。どちらにしても、きれいに磨きあげておくことで、いっそう心地よさが増してくるのは、おそうじ担当者だけの満足だろうか。

毛皮をまとったホンニャの夏はさぞたいへんだろうが、天然のクッションのような体は床にべったりとはりつくことができ、なんともうらやましい。人間ではそうはいかないし、材質のとりあわせも毛皮と床の対比にはかなわない。

湿度の高いこの午後、ホンニャは庭に近いリビングの床に、戸外をながめるポーズでよこたわっている。食卓の脚もとごしの、白く照り映える床に逆光のホンニャアがいて、静けさがあった。

しかし、彼のセンサーはけっして休むことはない。私がめくるかすかな紙の音や冷蔵庫のうなり声に、耳が小さく動き、しっぽが緊張する。まるで後頭部にも目があるようだ。

ふと、いたずら心がおこって、私はホンニャにさそいをかけてみる。

「ホンニャア、コロコロンは？」

庭を向いていたホンニャは反射的に上半身をひねって起こし、あたりの床をキョロキョロとみまわした。

「コロコロン」とは、ビー玉が床をころがる音の擬音なのだ。トオルが小学生のころ、床にビー玉をころがしてはホンニャアをじゃれさせて遊んだ。ホンニャ自身もビー玉との追いかっけは好きなようだったが、私たちがあまり楽しそうなので、いっしょうけんめいサーブスしているふうもあった。

「コロコロン」の言葉は、ビー玉をころがすたびに、「ニャアちゃん、コロコロン！」とくりかえしていたのを、いつのまにかおぼえたのだ。

もう何年も前の遊びを、ホンニャアがおぼえているだろうか？と試すつもりもあったのだが、「コロコロン」の情景は一瞬に彼のCPUからとびだしてきた。どこかの方向

あれってどこ置いたんだっけ？ っていうときは、自分がそれを置きそうなところや隠れそうなところをさがしますよね。ホンニャアにしても同じこと、長年培われた体験がさがしものにはモノをいうようです。

記憶のすき間

「コロコロン」の遊びを思いだしてしまったホンニャアは、のんびりと休むのはやめて、さがしものをはじめた。果物やワインの乗った赤いワゴンテーブルの下を、まずのぞいている。そうだ、以前はここにビー玉の入った小さな籐(とう)のカゴがあった。よくおぼえているものだ。あれをみつけたら、ビー玉遊びができると考えたのだろう。

子猫のころ、ポリエチレンの包装ひもでこしらえたボンボンが大好きで、遊びたくなると自分でくわえてきて、私たちの前にボンと投げだした。クールでわがままな彼だけれど、遊び以上に、私たちとの交流を望んでいるようすがしばしば感じられて、驚くことも多かった。

「コロコロンをさがしてるの？」

私はホンニャアに聞いてみた。

「ウン、どこにあるの？」という目で、ホンニャアは私を見上げる。

「どこかなあ」と、私はオーディオのラックのあたりをさがしてみせる。ホンニャアもイソイソと、私の横でいっしょにのぞきこむ。

夕飯をやるたび、「ゴハンゴハン」と語りかけていたら、とうとう猫が「ゴハンゴハン」と言うようになった話を聞いたばかりだったので、いまにそんなことが起こるかもしれない期待をかけて、ホンニャアと「会話」してみた。心がひとつになって、お互い意味することを伝えあえれば、それはき

っと会話と同じなのだ。

ところで、ビー玉はトオルの部屋にしまっているのだから、ホンニャアには申しわけないことになった。

「コロコロン、あるかな？」と、私はころがっているビー玉をさがすふりをして、カーテンの陰をのぞく。ホンニャアもついてきていっしょにカーテンの下に首を入れている。どうやってこの場をごまかそうかなと思っていると、ホンニャアはこんどは食器戸棚と冷蔵庫のすき間に小さな腕をつっこんでかきよせている。

細いすき間に腕のつけ根まで差し込んでいっばいに伸ばし、つかえた顔を横に向けて手の先に注意を集中しているようすがあまりにおかしい。

「あるわけないでしょ……」と思わず人間相手の調子で言いかけたとき、ホンニャアがこちら向きになって、同時にホコリまみれの丸いものがころがり出してきた。

「あらア……」と拾いあげてみると、それはビー玉よりはあまりに小さな、オモチャのガンにつめる弾丸だった。それでも、とりえずコロコロンの代替品をみつけ出したホンニャアに私は敬意をはらった。

ホンニャアは自分の記憶と経験から、ビー玉のたくわえられている本拠地をたしかめてみたり、それがころがって隠れこみそうなところをいくつか推理してみた。頭の中でじっさいにビー玉をころがして、第一の候補になったのが、冷蔵庫と食器戸棚のすき間だったのだ。

🐾さがし屋稼業

「サーチャー」という技術者が、このごろ注目を浴びはじめて、その資格を得ようとする人がどんどんふえているそう。

正確には「データベース検索技術者」といって、国内外のあらゆるデータベースから、必要な情報を引き出す専門家だ。基本的には、パソコン通信による各データベースへのアクセスと、必要事項の検索をするのだが、実務としての能力はなかなかたやすいものではないようだ。

「情報化社会」といまでも言われてきたものも、コンピュータの活用によって、この数年でますます過密になった。現在日本で利用できる商用データベースは、海外のものが1800あまり、国内は420ほどで、5年

間で4倍になったそう。

ある特定の「情報」を得たいと思ったとき、情報源が大きく豊富であるほど検索は複雑になり、そのための専門の知識を持った技術者が求められることになる。日本でも、そういった時代の要求から、代行検索業の会社がつぎつぎ誕生している。

そんなところで力を発揮しようという、躍進的ともいえる特殊技能のしごと、それが「サーチャー」だ。

サーチャーをめざす人のために、情報科学技術協会が昭和60年から毎年実施している、「データベース検索技術者認定試験」がある。この試験には1級と2級があって、まず2級をめざしてみんな勉強する。2級は「与えられた機器を使用して、なんとかひとりで適切な検索を行い得る能力をもつ人」(情報科学技術協会資料より)で、1級は「2級の延長上のより高度なランクであり、単に自分が適切な検索を行うことができるのみならず、初心者、2級合格者を指導、管理できる能力を持つ人」(同)だそう。

このサーチャーになりたい人というのが、前記の資料によると、5年前の受験者は223人、うち合格者140人、合格者のうち女性は45%。昨年度は受験者816人、合格者301人、同じく女性58%で、女性の比率が大きくなってきている。

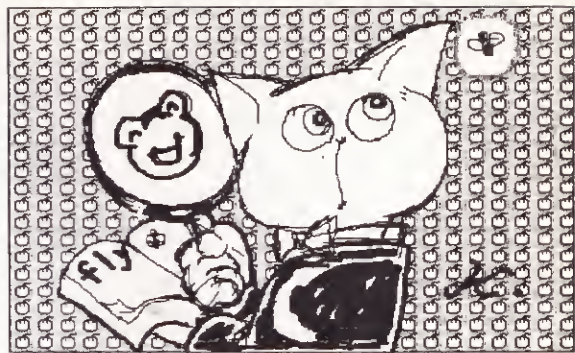
あるサーチャー講座の教室をのぞいたら、約35人の受講者のうち女性27人、男性8人で圧倒的に女性が多く、それもおおかたは20代だった。

単なるオフィスのオペレータとはちがう特殊な技術のいる職業として、なかなか魅力はあるものの、ただのカッコよさを求めているのでは少し甘いかと感じる。

🐾コロコロンの心

認定試験の問題もなかなかのむずかしさで、パソコン通信の知識はもとより、検索のための特殊なきまり、専門用語、略語の解釈が山のようにある。その上、暗記しなければならない、あまりにたくさんのデータベースの種類、名称、特色。

空欄をうめる問題では、たとえば内容は



情報検索についての一般論であっても、同義語、類似語の微妙な判別がとてむずかしい。試験問題そのものが、検索者としての推理や分解や組み立ての能力をためしているようだ。

とはいえ、試験は正解の数が多ければいいのだ。若い人ほど暗記力はすぐれているし、「合格」はなんとかできるかもしれない。だが、そのあとの実務の世界は、マシンをあやつるだけの知識では、たやすく成り立たないらしい。

もちろんいちばんものをいうのは、各データベースの内容、特色を、自分の頭の中のデータベースにいかにかくさん取り揃えているかということかもしれない。しかし目的は、依頼者の要求にいかにか適切に答えるかだ。

要求している人の目的や意図をじゅうぶんに理解する能力、その目的のために、どういう手順で検索をすすめていくかを組み立てる力。検索はかならずしもデータベースから始まるとは限らないそう。ときには、それ以前に「要求された情報」に関する分野の、専門家の意見が必要になることもある。そういった知己を持っていることも、検索技術者の力の一部だという。

そして、広い範囲で知識が豊かで、経験も多いこと。なによりも、インスピレーションが鋭くはたらくこと。この直感がサーチャーの腕を左右し、海外データベースへのアクセス時間も最小限にしてくれることだろう。料金も重要な条件だ。

こうしてみると、サーチャーというしごととは、人とコンピュータ、それぞれの本質を深く理解できなければつとまらない、なかなかやりがいのある新職業だ。そして、そのスピリットは、なんといっても「コロコロン」をさがし出したホンニャアのあのインスピレーションだ。

[第3話]

旅行あれこれ

TAKAHARA HIDEKI 高原 秀己

このところ、すっかりと海外旅行ブームは定着してしまい、もうブームなどと呼ぶのはふさわしくない。とくに年末年始や夏休みともなると、恒例行事といってもいいほどだ。日本人が海外の旅先で消費してくるお金は年間10億ドルだというからものすごい。

海外旅行にもいろいろな形式があるが、やはりパックスツアーが一番の動員力を誇っているようだ。北海道や沖縄に行くのとさして変わらない金額で海外の人気旅行地に行けてしまうのだから、人気が出るはずだ。いよいよ夏休み。

出不精のぼくも、せっかくの夏休みにも何もしないのももったいないので、旅行代理店に足を運んで調べてみたのだが、パックスツアーはさすがに安い。東南アジアやハワイ、グアムや東南アジアで10万円前後。15万円ちょっと出せば、アメリカ西海岸でもオーストラリアでも行けてしまう。

ところがいざ申し込んでみようとするとなかなか難しい。

「じゃあ、この12万8000円でオーストラリアっていうの、ありますか？」

「いっぱいですね。夏休みのピークの時期のは早めになくなってしまいますよ」

それで作戦を変更して夏休みをやや外してみることにしたのだが、それもなかなかうまくいかないようだ。

「8月末出発のシンガポール・マレーシア14万8000円っていうのはどうですか？」

「まだご予約がありませんね、何人様ですか？」

「ぼくだけです」

「それはどうですかねえ。おひとりですとツアーとして成立しませんので。他のお客様の申し込みを待って、ということになりますが、ご予約だけされますか？」

というわけで、旅行大作戦はひとまず延期することにして、旅行代理店から逃げ出してきた。

そもそも自分がカップルのひとりでないことが問題なのかもしれないのだが、それを気にしてはミジメになる。旅行代理店とパックスツアーのシステムが、いや社会全体

の歯車が狂っていることにして一件落着としてしまったのだが、この分では夏休みは今年もたいしたことはできそうにない。

どうも男性がひとりではぶらりと海外旅行をするっていうのは絵にならないようだ。そもそもがあまり、そういった不気味な客は想定されていないのだろう。

確かに雑誌でやっている旅行の特集企画にしても、ほとんど全部が女性向け。ある女性誌などは人気旅行地を毎号特集することに編集方針を変えてしまったほどだ。女性向け雑誌にはなくてはならないアイテムとなっている。

人気小説のトラベルミステリーなどにしても、たいてい事件を起こす客は女性かアベックと相場が決まっている。ひとり旅をする男性というのは刑事が探偵、あるいは出張しているビジネスマンと相場が決まっている。

かくいうぼくも、最近の旅行はスキーを除けば出張ばかり。

つい先日、九州を数日かけて回ってきた。久々に3日以上長さで、旅行らしい旅行だった。

仕事とはいえ地方に行くと、緑と青の自然の景色を満喫できるので、なかなかの気晴らしになる。なんせ日頃は緑といえばゴルフ場くらいしか緑のない生活を続けているのだから。

今回はキーボードから離れた生活をしたかったので、昨年末のアメリカ旅行で移動端末機として大活躍してくれたラップトップパソコン(NECの4kgのマシン)はあえて持っていかなかった。

もっともヘッドホンステレオとゲームボーイはしっかりと持っていった。この2つは退屈な飛行機や列車の中では欠かすことができない小道具だ。

九州旅行での訪問先のひとつはA社の地方工場。そこに勤務する、ある課長さんと飲みに出かけた。

その課長さん、もともとは東京本社勤務の人なのだが、ここ数年は地方工場を転々としているそうなのだ。

アルコールが十分回ってきた頃、彼はと

ても面白い話をしはじめた。

「妻がいうんですよ。私はA社という企業社会の中で生活しているだけなんだから。東京本社であろうが、地方工場であろうがそれほどの違いはない。ところが自分はその地域の中で生活しなきゃいけないんだから、転勤があると影響をモロに受けてしまう。だから嫌だってね」

これは盲点だった。

地方工場というのは、ロケーションこそたまたま地方にあるとはいえ、その企業の完全な一部分となって機能している。空間も工場という形で隔離されており、内部は企業社会の延長線上にある。

そこで働く人たちは県民とか町民という共通項でくくられているわけではなく、企業という名のパラレルワールドの住民なのだ。だから地方にいても、実際には地方で生活していることにはならない。

これは外資系企業のIBMとかTI(テキサス・インスツルメンツ)、インテルとかを考えてみると、さらにわかりやすい。

建物のデザインや内装からして、しっかりとそれぞれの企業カラーが打ち出されている。内部での生活様式ならぬビジネス様式も統一されている。

入り口を通り抜ければ、もう六本木の本社の中にいるのか、地方工場にいるのかすらはつきりしないほどだ。アメリカの本社ですら、違和感はない。

これからは企業が人々の生活に占めるウエイトがますます高まってきて、国や地域の差を吸収していくという説がある。

実際にこうした地方工場の機能を見ると、日本企業に限らず、国家とか自治体という縦割りの社会よりも強力な横割りの企業社会がジワジワと浸透してきているような気がする。

これについていける人についていけない人とは大きな違いが出てくるのだろう。

ちなみにその課長さん、さすがに3回目の転勤とあって、家族は東京近郊の家に残ってしまい、哀れ単身赴任となっているそう。彼がいつ東京本社に戻れるのか、まったく彼にもわからないようだ。

BACK ISSUES

バックナンバー案内

ここには1989年8月号から1990年7月号までをご紹介します。現在1989年7〜12、1990年1〜7月号までの在庫がございます。バックナンバーおよび定期購読のお申し込み方法については、176ページを参照してください。

1989



8月号

特集1 X1プログラミングガイドブック
PCGの基礎から奥義まで/超高速ラインルーチン 他

特集2 3Dグラフィックの深淵へ
スキャンラインバッファ/3Dモデリング 他

新連載 (で)のショートプロバレー

X68000マシン語プログラミング/C調言語講座 PRO-68K
X-BASICプログラミング調理実習/D6GA-CGA講座
MZ-2500用グラフィックエディタ/Z80's Bar 他
全機種共通システム CP/M用ファイルコンバータ



9月号

特集 活用ハードディスク&プリンタ

各社ハードディスク接続総チェック/ハードディスク雑学講座/COPYキーメニュー/ビデオプリンタ活用プログラム 他

THE SOFTOUCH ジェノサイド/琉球/mFORTH Compiler
●サイバースティックで遊ぶ 不思議な環境ソフトの世界

●X1/X1turbo用シューティングゲーム Defeat X
Z80's Bar/MZ-2500グラフィックエディタ 他

[X68000] X-BASIC/マシン語/C調言語講座/D6GA-CGA
全機種共通システム 生物進化シミュレーションBUGS



10月号

特集 ゲーム面白心理学

ソーサリアン・宇宙からの訪問者/ファンタジーゾーン
ねじ式/ガウディ・バルセロナの風/サバッシュ 他

●MZ-700用シューティングゲームSide Roll-F
●X1/X1turbo用カードゲームBonding

ショートプロ/Z80's Bar/MZ-2500グラフィックエディタ
X68000マシン語/X-BASIC/C調言語講座/D6GA-CGA
THE SOFTOUCH Z'sTRIPHONY DIGITAL CRAFT/James68K
全機種共通システム 小型インタプリタ言語TTI



11月号

特集 microComputer入門

初歩からのCPU物語/RISCプロセッサの設計と製作
X68000&X1で周辺LSIを使いこなそう

連載 ショートプロ/Z80's Bar/MZ-2500グラフィックエディタ
X68000マシン語/X-BASIC/C調言語講座/D6GA-CGA

●X68000用カードゲームばばぬき

LIVE in '89 メタルホーク/オブ・ラ・ディ、オブ・ラ・ダ
THE SOFTOUCH Stationery PRO-68K/リングマスター1
全機種共通システム TTI用パズルゲームPUSH BON!



12月号

特集 Cプログラミングへの招待

付録 C言語簡易リファレンス

連載 ショートプロバレー/Z80's Bar

X68000マシン語/X-BASIC/D6GA-CGA

●Oh! X2周年特別企画「素粒子の音が聞こえる」

●X1/turbo用アクションゲームACTIVE UNIT

LIVE in '89 天空の城ラピュタ/ギャラクシーフォース
THE SOFTOUCH 38万キロの虚空/た〜みのる2
全機種共通システム SLANG用リダイレクションライブラリ

1990



1月号

特集1 オペレーティングスタイルの研究

特集2 Cプログラミング応用編

連載 ショートプロバレー/Z80's Bar

X68000マシン語/C調言語講座/D6GA-CGA

●X1/turbo用シミュレーションゲームSuper Battle

LIVE in '90 さよならを過ぎて/RIDEEN

THE SOFTOUCH レナム/メタルサイト
全機種共通システム WORM KUN/再掲載SLANG
特別付録 X68000 THE SOFTWARE CATALOGUE



2月号

特集 画像圧縮へのアプローチ

連載 ショートプロバレー/Z80's Bar/D6GA-CGA

X68000マシン語/C調言語講座/X-BASIC調理実習

●X68000用ゲームプログラムGon Gon

●MZ-700用紙芝居Eyelarth

LIVE in '90 オーダイン/魔女の宅急便

THE SOFTOUCH A-JAX/フラッピー2/夢幻戦士ヴァリスII

マジックパレット/Mu-1/CYBERNOTE PRO-68K

全機種共通システム 超小型コンパイラTTC++



3月号

特集 MUSICアドベンチャー

X68000用MIDIドライバ&音源エディタ

なんでも鳴らせるOPMD.X/MMLを楽譜データに

連載 ショートプロバレー/Z80's Bar/D6GA-CGA

C調言語講座/X-BASIC調理実習

●X1/turboシミュレーションCRISIS in Tokyo

LIVE in '90 パワードリフト/スキーム/となりのトロ

THE SOFTOUCH ナイトアームズ/斬/ダンジョンマスター

全機種共通システム 超多機能アセンブラOHM-Z80



4月号

特集 ゲームシステム文学誌

1989年度GAME OF THE YEAR発表

連載 ショートプロバレー/Z80's Bar/D6GA-CGA

X-BASIC調理実習/C調言語講座/X68000マシン語

●X1-MZ-2000/2500用RPG The Cave of Dalk

●うわさの68040, ついに登場

LIVE in '90 パーニングフォース(OPMD対応)

THE SOFTOUCH The Fille Professor/HOST PRO-68K

全機種共通システム ファジコンコンピュータシミュレータ-MY



5月号

特集 BASICプログラミング

第5回 言わせてくれなくちゃだわ

連載 ショートプロバレー/Z80's Bar

X-BASIC調理実習/X68000マシン語プログラミング

●新機種X68000SUPER-HD/EXPERTII/PROII

●ラジコンスティックの製作

LIVE in '90 TURBO OUTRUN

THE SOFTOUCH 天下統一/ボビュラス/Hyperword

全機種共通システム インタプリタ言語STACK



6月号

特集 創刊8周年記念PRO-68K(付録5"2HD)

Oh! Xアンケート結果大分析大会

連載 ショートプロバレー/Z80's Bar/PurePASCAL

X-BASIC調理実習/X68000マシン語プログラミング

●X1/turbo用コマンドシェルシミュレータ

●ハードウェア工作入門

LIVE in '90 ナイトアームズ/悪魔城伝説/この木なんの木

THE SOFTOUCH 三国志II/FAR SIDE MOON/グラナダ

全機種共通システム X68000用S-OS"SWORD"他



7月号

特集 マシン語への第一歩

X68000SUPER-HD試用レポート

連載 ショートプロバレー/Z80's Bar/D6GA-CGA

X-BASIC調理実習/PurePASCAL

●INTEGRAL XI——ノーマルXIへの対応

●ハードウェア工作入門

LIVE in '90 夢幻戦士ヴァリスII/トッカータとフーガ二短調

THE SOFTOUCH サークあーくしゅ/ダウンタウン熱血物語

全機種共通システム リロケータブルアセンブラWZD

NEW PRODUCTS

スーパーアウトラインフォント内蔵

WD-A320/340

シャープ



WD-A340

シャープは「見やすい大型液晶画面」、「活字に迫る高品位印刷」、「思いどおりのレイアウト」、「正しいことばづかい」などを追求したラップトップ型ワープロ「WD-A320」および「WD-A340」を発売した。

「WD-A320/340」は新開発の専用LSIにより名刺用の小さな文字から拡大文字まで美しくなめらかに印字する、「書院スーパーアウトラインフォント」を内蔵している。曲線データで文字を形成しているため、直線（ベクトル）データによるアウトラインフォントに比べ品位を向上している。4.5～288ポイントまで合計67種類のマルチポイント文字（欧文時はマルチポイント23種類）を自由に設定することで、多彩な大きさの文字を利用できる。また、それに加えて64ドット・400DPIの高精細プリンタを搭載していることで、美しい印字が可能となっている。

さらに、パーソナルDTP機能、手紙文の作成に便利な「直子の代筆（書院版）」、15万例のAI-V3辞書、電子手帳とのデータの共有ができる電子手帳機能などの機能も装備している。

「WD-A340」ではこれに加えてハイコン

トラスト白黒液晶画面、類語辞書、文体統一機能などの文書校正支援機能、MS-DOSコンバータ、通信ソフトなどを搭載している。価格はそれぞれ178,000円と198,000円（どちらも税別）。

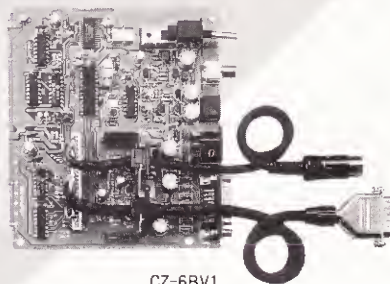
〈問い合わせ先〉

シャープ(株) ☎06(621)1221,03(260)1161

X68000用ビデオボード

CZ-6BV1

シャープ



CZ-6BV1

シャープはX68000用の周辺機器としてビデオボード「CZ-6BV1」を発売した。このボードをX68000の拡張I/Oスロット（2スロット分を使用）に装着することにより、コンピュータ映像をビデオ信号として取り出すことができるようになる。たとえば、X68000上で作ったグラフィックやアニメーションあるいはゲーム画面などを手軽にVTRに録画することができる。さらに、ビデオ入力端子のついている液晶ビジョンや大型テレビにX68000を接続して、迫力ある大画面でゲームなどを楽しむこともできるようになる。特徴は以下のとおり。

- ・NTSCエンコーダ、同期信号発生回路とも1チップ化
- ・入出力端子は以下のものを装備
アナログRGB×2
テレビコントロール×2
S映像出力×1
コンポジットビデオ出力×1

・高解像度モード時のビデオ出力を自動的に停止することができる

価格は21,000円（税別）。

〈問い合わせ先〉

シャープ(株) ☎06(621)1221,03(260)1161

X68000とMacをリンク

Mac版「XIN/XOUT」

電機本舗

電機本舗はRS-232Cを介してデータ転送をするシステム、「XIN/XOUT」のMacintosh版を発売した。これはRS-232C/422通信ポートを利用して、Macintosh Plus, SE, SE/30, IIとMS-DOSマシン/X68000の間でのファイル転送を可能にするものである。バイナリファイルの転送も可能で（エラーチェックは独自のものを採用）、ファイルの一括指定一括転送もサポートしている。転送に際しては、転送先のファイル形式に自動変換、OSの相違を完全吸収し漢字を含んだファイルも正確に転送する。英語、日本語環境およびマルチファインダ上にて動作する。

パッケージにはRS-232Cケーブルと、ファイル転送プログラムのMac版とMS-DOS（/X68000/PC-DOS）版のフロッピーディスク2枚が入っている。価格は12,800円（税別）。

〈問い合わせ先〉

(有)電機本舗 ☎03(447)1773, BBS

03(447)2564 1200bps

XIN/XOUT



電子手帳用プリンタ&名刺管理カード
CE-80P, PA-7C50/7C51
シャープ



シャープは既存の電子手帳すべてに接続可能なプリンタ「CE-80P」を発売した。さらに、面倒な名刺の整理に便利な名刺管理カード「PA-7C50/51」を7月25日に発売する。

電子手帳用プリンタ「CE-80P」にははがきやラベルへの宛名印字はもちろん、リフィルへの住所録印字もできる。別売のはがきフィーダを装置すれば、連続20枚までのはがき裏面の連続印字が可能。年賀状などで使うあいさつの慣用句73種類を内蔵しており、また、オプションの毛筆体カートリッジ「CE-61M」により美しい毛筆体での印字が可能になるので年賀状などが簡単に作成できる。リボンカセットは黒、赤、青、茶、金、銀が用意されていて（茶は8月発売予定）、6色印字が可能。価格は45,000円（税別）。

名刺管理カード「PA-7C50/51」は名刺情報はもちろん、いつ、どんな用件で会ったのかを記憶できる交際録、趣味や嗜好を記憶できる備考、年賀状やお歳暮などの状況をチェックできるチェックリストなどの記憶が可能。名刺情報は名前4文字、電話番号12桁、FAX番号12桁、会社名8文字、所属5文字、役職2文字、郵便番号3桁、住所20文字の場合で約350人分（PA-7C50の場合は約160人分）が記憶できる。機能としては郵便番号辞書、日付検索やチェック検索などの多彩な検索機能、宛名印字機能を搭載。さらに本体メモリをバックアップできるRAMファイルとしての使用も可能

となっている。価格は「PA-7C50」が13,000円、「PA-7C51」が16,000円。

〈問い合わせ先〉
シャープ㈱ ☎06(621)1221, 03(260)1161

32ビット浮動小数点DSP
DSP96002
モトローラ

モトローラは24ビット固定小数点デジタルシグナルプロセッサDSP56000ファミリの上位機種として、32ビット浮動小数点DSP96002を開発した。

- ・動作周波数：27MHz, 33MHz
- ・命令サイクル：74nsec, 60nsec
- ・IEEE754データフォーマットに準拠
- ・43×43ビット→96ビット浮動小数点演算
- ・32×32ビット→64ビット整数演算
- ・12Gワードのメモリ空間
- ・1KワードのオンチップデータRAM
- ・1KワードのオンチップデータROM（サイン、コサインテーブル）
- ・512ワードのオンチッププログラムRAM

- ・2チャンネルDMAC
- ・32ビットバレルシフタ
- ・223ピンセラミックPGAパッケージ
- ・割り算と平方根用に高速な命令（6命令サイクルと9命令サイクル）を用意

DSP96002の2つの外部メモリ拡張ポート（ポートAおよびポートB）はユーザープログラミングによって、外部メモリのアクセスポートあるいはホストプロセッサとの接続ポートとして使用できる。さらに、DSP96002の各ポートにはマルチプロセッサ構成をサポートする信号線も用意されているので、複数のDSP96002でマルチプロセッサを構成し高性能な演算処理を実現することもできる。

以上のような特長により、DSP96002は従来のDSPでは処理が困難であった画像処理、浮動小数点演算アクセラレータ、医用機器、周波数解析処理などに応用が可能である。

〈問い合わせ先〉
モトローラ㈱ ☎0120-068030

I N F O R M A T I O N 番外編

「X68000グッズショップ in Akihabara」

ミナミ電気株式会社 本館5階

X68000グッズが買いたいと思っても、いままでは常備店がなかったので、イベントに行っても買うなどしか方法がありませんでした。しかし、このたびミナミ電気本館5階のパソコンフロアにX68000グッズショップ in Akihabaraが開設されることになり、いつでもX68000グッズを手に入れることができるようになりました。

そこで、それを記念してひょっとしたらあまり知られていないかもしれないグッズの数々を紹介してみたいと思います。

- ★X68000牛革ベルト
標準価格6,300円（税別）
バックルには光輝く「X」のロゴが……
- ★X68000キーホルダー
標準価格1,300円（税別）
X68000の電源スイッチにも鍵があればよかったのに
- ★X68000ネクタイピン
標準価格3,000円（税別）
ネクタイをする人にはいいかも

- ★X68000電飾POP
標準価格9,500円（税別）
暗い所で見ると本当にきれい
 - ★X68000クリスタルボルシェ
標準価格8,000円（税別）
ガラスでできたボルシェ911
 - ★X68000ジッポ・ライター
標準価格4,800円（税別）
あのツタンカーメンの仮面が……
 - さらに、
 - ★X68000ゴルフボール
標準価格1,900円（税別）
 - ★X68000傘
標準価格4,200円（税別）
 - ★X68000スポーツタオル
標準価格3,300円（税別）
- と、「こんなものまで？」と思うような変わった(?)商品が、ほかにまだまだいろいろあります。興味のある方はお店でご覧になるとよいでしょう。
- ☆万世橋交差点際 第一家電隣



牛革ベルト

キーホルダー/タイピン

電飾POP

ジッポ・ライター

ゴルフボール

FILES Oh!

このインデックスは、タイトル、注記——
筆者名、誌名、月号、ページで構成されて
います。毎日暑い日が続きますね。夏バテ
や寝冷えに気をつけて、楽しく有意義な夏
休みを過ごしてください。

一般

▶特集シムアース

シムシティの登場によって示されたパソコンシミュレーションの楽しさ。今度はもっとグローバルに地球環境のシミュレーションをやってみよう。そこで発表されたのが「シムアース」。その概念や裏話などを解説。シムアースを考える座談会にはミュージシャンの細野晴臣、戸田誠司、日本自然保護協会の横山隆一らが参加している。——編集部、LOGIN、12号、116-127pp.

▶ネットワーク・ホリック 第22回

新聞の申し込みまでできちゃうぞ。大手ネットのショッピングサービスを紹介。PDSはPC-9801のZMODEM転送プログラム「ZM.EXE」、X68000のシューティングゲーム「MEMORY BROKEN.X」。全国BBS探訪記は秋葉原にあるPENCIL-NET。——編集部、LOGIN、12号、202-203pp.

▶ハードラボラトリー

MIDIについて解説。X68000の純正MIDIボードCZ-6BMIやMusicstudio PRO-68Kも紹介。——編集部、POPCOM、7月号、106-108pp.

▶X68000のウイルス騒動の真相

先頃新聞を騒がせたX68000用市販ソフトへのウイルス混入事件についてウイルス騒動の当事者が内情を語る。日コン連では昨年11月に各マスコミへ今回のウイルスのソースリストを送っていたという。——日コン連理事長山本隆雄、The BASIC、7月号、176-177pp.

▶2大ショウに見る最新パソコンの現状

ビジネスショウ・マイコンショウに展示された各社の新製品をレポートし、今年のトレンドを探る。——編集部、マイコン、7月号、135-144pp.

▶コンピュータ・ウイルスを考える

ウイルスについて正しい理解をするために、ウイルスの種類や事例、対策について述べる。——コンピュータ・ウイルス研究会、マイコン、7月号、164-165pp.

▶楽器が弾けなくても、声で楽器が演奏できる

マイクロコンピュータショウに展示されていた、ボイスインプットを紹介。マイクに入力された音程を解析してMIDI楽器を鳴らすことができる。——FORESIGHT企画部・藤本健、マイコン、7月号、239-240pp.

▶ビジネスマンの情報管理術

害者のヨーロッパ旅行記第3弾。ポルトガル、オランダ、イギリスなどで7カ国語翻訳カードと通貨換算機能が活躍する。——塚田洋一、マイコン、7月号、310-312pp.

▶やまさんのアルゴリズム・ブック

MS-DOSなどで頻繁に使われるワイルドカード機能のアルゴリズムを考える。——やまさん、マイコン、7月号、321-325pp.

▶実践ハード入門

梅雨にあわせて、湿度センサを使った簡易湿度計を作る。——石川至知、マイコン、7月号、334-336pp.

▶レーザーディスクで広がるマルチメディアの世界

レーザーディスクの生み出すハイパーメディアの世界について述べ、またマッキントッシュでのハイパーメディアの現状を報告する。——田島恵介・長谷川昌夫、マイコン、7月号、346-354pp.

▶NEW MACHINES '90

NEC、エプソンなどの新機種と共に、AX仕様のAll in Note、X68000SUPER-HDを取り上げ、概要を紹介する。——編集部、ASCII、7月号、258-280pp.

▶AtariSTの魅惑の世界

68000使用のホビーパソコン、米Atari社のSTシリーズの魅力に迫る。今月はラインナップ、ハードウェア、PDSやゲーム事情などについて。——小沢靖・池田賢司・判治聡、ASCII、7月号、313-320pp.

▶MEDIA BREAK

北九州市八幡にオープンしたスペースワールドの宇宙飛行士訓練プログラム「スペースキャンプ」を紹介。——浦山明俊・佐藤守弘、ASCII、7月号、409-411pp.

MZシリーズ

MZ-1500 (MZ-5Z001 BASIC)

▶1582

カプコンのシューティングじゃないよ。戦国アクションゲーム。——大石豊、マイコンBASIC Magazine、7月号、126-128pp.

MZ-2500 (BASIC-M25)

▶BLOCK BROKEN

ブロックと入れ替わる難解パズルゲーム。——Tak KuN、マイコンBASIC Magazine、7月号、129-130pp.

▶Multi Window

BASICのウィンドウサブルーチン。——佐藤拓也、マイコンBASIC Magazine、7月号、179-180pp.

X1/turbo/Z

X1シリーズ

▶最新ゲーム徹底解剖!!

新着ゲーム「スライミヤー」の基礎攻略法を紹介。——編集部、LOGIN、11号、226-227pp.

▶攻略おすすりめゲーム

ウィザードリィVの地下3階までを攻略。——編集部、テクノポリス、7月号、50-53pp.

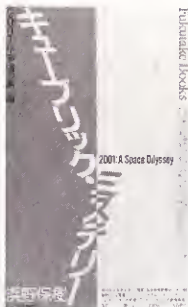
▶桃四郎

好評の桃シリーズ、今回は桃太郎4人目の兄弟の話。お供をやとい鬼をたおすアクションゲーム。ジョイステ

参考文献

I/O 工学社
ASCII アスキー
コンプティーク 角川書店
The BASIC 技術評論社
テクノポリス 徳間書店
POPCOM 小学館
マイコン 電波新聞社
マイコンBASIC Magazine 電波新聞社
LOGIN アスキー

新刊書案内



この人の著書(「ハイパーメディア・ギャラクシー」など)を読むと、実に「2001年宇宙の旅」に関する話が多い。趣味が高じてか今度は「2001年宇宙の旅」を中心においた映像論の本を書いた。本書は2つの点で実に面白い。ひとつは、そこいらの映画評論家が書く映画評より資料も視点もしっかりしていること。もうひとつは、どうして著者はコンピュータはメディアを目指すべきだと考えるのか。メディアとなったコンピュータに何を期待するのかがはっきりとわかることだ。

オーソン・ウェルズ、小津安二郎、そしてキューブリックの3人の映画監督の共通点。彼らは何

と戦い、何を表現しようとしたのかということ。「ジョージ・ルーカスやスティーブン・スピルバーグは、最新の特撮技術を総動員して、過去のイメージを増幅しているだけ」だということ。HALはなぜ殺さねばならなかったのかということ(2010年で示されたような安易な答えではない)。「2001年宇宙の旅」はメディア論だということ。著者はメディアとしてコンピュータを使うことによって、個人の表現を復権させたいのである。(K) キューブリック・ミステリー 浜野保樹著 福武書店

☎03(230)2131 新書判 204ページ 1,130円

イック専用。——ズオ、マイコンBASIC Magazine, 7月号, 158-160pp.

▶LEADER LEADER

シルクハットをかぶったハット君にパンを食べさせてゴールに向かう。風船で道をつくってハット君を誘導する。風船バズルゲーム。——吉川章、マイコンBASIC Magazine, 7月号, 161-164pp.

▶性格判断

学園祭の定番、性格判断プログラム。多少判定の文章が貧しいという声もなくはないが……。——編集部、マイコン, 7月号, 212-216pp.

X1+FM音源ボード(要NEW FM音源ドライバ)

▶ミスティ・ブルー

エニックスのアドベンチャーゲームのミュージックプログラム。——KENJI, マイコンBASIC Magazine, 7月号, 192-194pp.

X1 turboシリーズ

▶NEW SOFT

セレクトッドソーサリアン4のシナリオの解説。——編集部, LOGIN, 12号, 12-13pp.

▶攻略おすすめゲーム

世界の海を股にかけるゲーム、「大航海時代」の最も重要な要素、交易について攻略。——編集部, テクノポリス, 7月号, 46-49pp.

▶月に帰りたいヒトデちゃん

降ってくる星を足場にして月まで帰る。スクロールアクションゲーム。——HARU, マイコンBASIC Magazine, 7月号, 164-165pp.

X68000

▶NEW SOFT

7月発売予定の「ウルティマV」と「闇の血族」、そのほか発売中の「バズニック」「天下統一」「ダウタウン熱血物語」を紹介。——編集部, LOGIN, 11号, 12-25pp.

▶X68000新聞

戦国ゲーム特集。「天下統一」をはじめ「信長の野望・全国版/戦国群雄伝」「斬(ZAN)」を紹介。そのほか「POOL BAR」「闇の血族」「ダウタウン熱血物語」「ガンシップ」を紹介。——編集部, LOGIN, 11号, 162-167pp.

▶最新ゲーム徹底解剖!!

新着アクションゲーム「グラナダ」の攻略・その2。ステージ4からステージ6までを、マップを載せて紹介。アクションパズルゲーム「スライミヤ」も紹介。——編集部, LOGIN, 11号, 196-199・226-227pp.

▶Software Review

ポピュラスを真面目に考えてみる! ほかのゲームとはちょっと違うポピュラスの面白さは? ——川村B, LOGIN, 11号, 230-231pp.

▶NEW SOFT

8月発売予定のシミュレーションゲーム「JOSHUA」、7月発売予定の「POOL BAR」を紹介。——編集部, LOGIN, 12号, 19・22p.

▶X68000新聞

新着ゲームの紹介。「ラグーン」「維新の嵐」「ルーンワース」。そのほかジェノサイドのCDレコーディング風景やThe File Professorの解説。——編集部, LOGIN, 12号, 130-135pp.

▶先取りおすすめゲーム

7月中旬発売予定の「ラグーン」を紹介。——編集部, テクノポリス, 7月号, 14-15pp.

▶GAMING WORLD

好評のくにおくんシリーズ「ダウタウン熱血物語」、アクションパズルゲーム「バズニック」「スライミヤ」「タッグ・オブ・ウォー」、発売予定の「ユニオン」「レインフォーサー」「RYU〜哭きの竜より〜」を紹介。——編集部, テクノポリス, 7月号, 18-30pp.

▶攻略おすすめゲーム

第二次大戦のフランス戦をあつかった陸戦シミュレーションゲーム「機甲師団」を徹底攻略。——編集部, テクノポリス, 7月号, 56-57pp.

▶レモンちっくWORLD

発売予定の美少女RPG「ランス2〜叛逆の少女たち〜」、麻雀ゲーム「びんびん麻雀ビーチエンゼル」、カードゲーム「DOKI DOKI Card League」を紹介。——編集部, テクノポリス, 7月号, 72-79pp.

▶SLGの夏が来た!!

シミュレーションゲーム特集。ポピュラスの紹介やその原作者ビーター氏からのありがたいお告げなど。——編集部, POPCOM, 7月号, 62-63pp.

▶WE ARE THE X68000 WORLD IN HOKKAIDO

新着ゲーム「ラグーン」「POOL BAR」「Vessel」「サーク」「ルーンワース」「レインフォーサー」「ユニオン」などスプライトツール「びくせる君」を紹介。——編集部, POPCOM, 7月号, 68-72pp.

▶ゲームがオレを呼んでいる!

くにおくんシリーズ「ダウタウン熱血物語」と発売予定のゲーム「ウルティマV」の攻略法を解説。——編集部, POPCOM, 7月号, 82-90pp.

▶バズルDEバトル

新着パズルゲーム「バズニック」を紹介している。——さすらいのバズラー, POPCOM, 7月号, 92-93pp.

▶ミュージックパビリオン

映画「香港パラダイス」の主題歌「無敵のビーナス」(GO-BANG'S)のミュージックプログラム。——編集部, POPCOM, 7月号, 176-179pp.

▶キミのX68000を護れ!

コンピュータウイルスの基礎知識ほか, X68000のIPL,

SRAM常駐型ウイルスに対して有効なワクチンソフトを誌上公開。——GORRY, マイコンBASIC Magazine, 7月号, 67-73pp.

▶誌上公開質問状

X-BASICの画像フォーマット「GL3」の解説や、カラーイメージユニット「CZ-6VT1」の機能紹介。そのほかCommunication PRO-68KでATモデムは使えるか? などの質問に答えている。——多田太郎, マイコンBASIC Magazine, 7月号, 90p.

▶わかった!

画面に隠れたアルファベットを当てる。マウス専用、文字がしゲーム。——小野正明, マイコンBASIC Magazine, 7月号, 166-167pp.

▶PYRAMID BREAK

ピラミッド型につまめた5種類のブロックを落とさずにとりていく。山くずしゲーム。——高道賢治, マイコンBASIC Magazine, 7月号, 168-170pp.

▶リレーレビュー

ウルフ・チームの「グラナダ」について、4人のライターの見解を聞く。——編集部, マイコン, 7月号, 194-195pp.

▶スクリーンエディタEDX

Human68kとOS-9/X68000上で共通の操作環境を提供するスクリーンエディタ。いわばEDXの機能強化版である。——村田誠, ASCII, 7月号, 335-338pp.

▶AV STRASSE

PDSのグラフィックエディタ, MFGEDを紹介。高機能ではないが瞬時に立ち上がる小回りの良さが身上。——仲田津弘, ASCII, 7月号, 353-356pp.

▶NEWBAT.X

以前発表されたBATKEY.Xのバージョンアップ版。パッケージの機能を拡張してくれる。——牛島健雄, I/O, 7月号, 198-202pp.

▶迷路エディタ

最大511×511のマス目にマウスで絵を描くと、それを正解として迷路を作ってくれるというもの。——カバウシ2世, I/O, 7月号, 189-197pp.

ポケコン

PC-E500

▶TURBO RUN

ドライビングゲーム。——森萬周作, マイコンBASIC Magazine, 7月号, 175p.

▶DRAGON BUSTERD

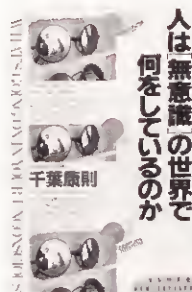
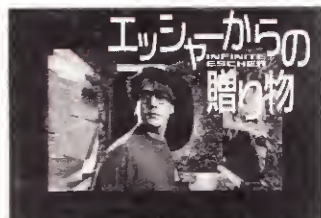
ドラゴンバスターことクローブスを操作してドラゴンをやっつける。アクションゲーム。——広鹿太一, マイコンBASIC Magazine, 7月号, 175-177pp.

エッシャーからの贈り物

エッシャーの描いた数々の作品を、CGで表現した。同じ内容のビデオも発売されており、そちらのほうがメインのようだ。作品の質としては今ひとつの感があるが、ビデオで見るとまた違った味わいだろう。エッシャーの騙し絵をCGにしちゃおうという発想はなかなかよい。(K)

野崎昭弘著 小学館

☎03(230)5442 B5判 47ページ
1,680円



人は「無意識」の世界で何をしているか

無意識の世界。カッコよくいうと、サブリミナルとか潜在意識とかとなる。本能や反射など、とにかく、人間のほとんどの活動は意識に現れないところで行われている。自分は意志に基づいてのみ行動していると思っている人、これを読んで謙虚になりなさい。PHPくさいところがわずかにあるが、丁寧な語り口で脳と無意識と行動の話を紹介している。専門的な内容はほとんどない。わからないことはわからないとしているのも善良。(K)

千葉康則著 PHP研究所

☎03(239)6221 B6判203ページ 1,000円



X68000のアセンブラで乱数発生プログラムを組もうと思うのですが、乱数発生原理がわからず困っています。乱数発生原理（乱数は1ロングワードの整数）はどうなっているのでしょうか？ 徳島県 森上 晶仁



一般に乱数は線形合同法と呼ばれる方法で作られています。これはある式に値を代入して計算によって乱数を生成する方法で、詳しい説明が1988年8月号に紹介されていますから興味のある方はそちらをどうぞ。

ところで、X68000には乱数を生成するためのファンクションコールが用意されていますから、それを利用することにして使い方を説明しましょう。

まず、このファンクションコールはFLO ATn.Xを組み込むことによって使えるようになるものです。乱数発生部のコール番号は\$FE0Eとなっていますのでアセンブラで書くなら、

```
dc.w $FE0E
```

もしくは、FEFUNCHをインクルードして、

```
FPACK RAND
```

(戻り値はd0.w)

という具合に使うことになります。

また、乱数系列の初期化には、

```
dc.w $FE0D
```

```
FPACK SRAND
```

(引数はd0.w)

とします。内容はBASICのRAND(), SRAND()と変わらないと思います(たぶん)。

ここで得ることのできる乱数の値の範囲は、0から32767と森上さんの希望とは違うものですが、実際には32ビットの乱数を必要とされることは稀だと思いますし、もし必要なときはこの方法で得た乱数にビットシフトなどの加工をしてから、さらに乱数を加えとか、工夫次第でどうにでもなるでしょう。



編集室の皆様こんにちは。僕は2年たってもろくにプログラム組めない大バカ野郎です。6

月号の付録のディスクはとてもよかったです。大事に使わせてもらっています。僕は前からCGをやってみたいと思っていました。だからANGELが動くのを楽しみにしていたのです。

いざ解凍してみてコマンドモードで“ANGEL”と入力してみると、「主記憶が足りません」と出てきました。ASK68Kをはずしてみなさいと書いてあったので、自分なりにはずしてみましたが同じメッセージしかでてきません。もう一度ASK68Kをはずすところからできるだけ詳しく書いてください。機種はX68000ACE, Human68k Ver.1.01, メインメモリは1Mバイトです。

愛知県 藤田 聡



同じ内容の質問がほかにも何通か送られてきましたが、藤田さんのハガキが一番最初に送られてきました(往復ハガキは使わないでくださいね)。とにかくX68000というマシンはメモリを大量に必要とするマシンです。標準で1Mバイトしか積んでいないマシンを使っている方は、BASICから子プロセスを実行することもままならないでしょう。

普通に考えれば、メモリを増やすにはパソコンショップにいった増設メモリを買ってこなくてはいいませんが、とりあえず使うことのないデバイスドライバを組み込まないようにしてメモリの空き容量を増やすことも可能です。質問電話によると藤田さんと同様のケースではほとんどがビジュアルシェルの起動のためのメモリ不足でした。このあたりの話は先月号でも触れていましたが、もう少し詳しく話しましょう。

Human68kは起動したドライブに存在するCONFIG.SYSの内容に従ってデバイスドライバの組み込みを行います。つまりASK68Kなどのデバイスドライバを組み込まないということは、CONFIG.SYSの内容を変更することにほかなりません。それにはエディタ、ワープロ、またはCUSTOM.Xのどれかを使うことになりますが、ここではエディタを使って変更するとしましょう。まず、

```
ED A: ¥CONFIG.SYS
```

としてCOFNIG.SYSをエディタに読み込みます。この場合はED.Xがパスの通っているディレクトリにあり、CONFIG.SYSがドライブAのルートディレクトリ上にあるものと考えています。画面のどこかに、

```
DEVICE=¥SYS¥ASK68K.SYS...
```

といった行があるはずですから、それを

```
*DEVICE=¥SYS¥ASK68K.SYS...
```

と先頭に*を挿入します(*をつけると注釈行扱いとなる)。こうしてからESC・Eでファイルをセーブしてエディタを終了させます。これでASK68Kを組み込まないシステムの完成です(注:リセットして再起動しなくてははいけません)。

ほかにも登録したくないデバイスドライバがあったら、同様の変更をすることで組み込まないようにすることができます。プリンタドライバやPCMドライバもとりにあえざらないでしようし、間違ってもRAMディスクを設定してはいけません。

また、Human68k Ver.2.0などには、

```
OPMDRV.X
```

```
HISTORY.X
```

```
FLOATn.X
```

```
IOCS.X
```

など、実行可能ファイルのくせにデバイスドライバとして登録できるものがあります(このうち、必ず設定しなければならないのはFLOATn.Xのみです)。FM音源を使うならOPMDRV.Xをデバイスドライバとして登録するために、

```
DEVICE=OPMDRV.X
```

と書くことになっていますが、そうしなくともコマンドモードから、

```
A: ¥SYS¥OPMDRV
```

とすれば、FM音源を使うことができますし、

```
A: ¥SYS¥OPMDRV OFF
```

とすれば、いつでもFM音源を使わないようにすることができます(使えなくなるだけで空きメモリが増えるわけではない)。

ですからFM音源を使うことが減多にないのなら、OPMDRV.Xを組み込まないようにしたほうがいいでしょう(標準1Mバイトの方は特に)。OPMDRV.Xを使用する

ソフトを起動したときは、エラー(\$FE0D)が発生しますから、そしたらOPMDRVとコマンドモードから入力すればいいのです。こうしておけば、OPMDRV.Xを使わない場合は通常87000バイト、コマンドモードから登録した場合も、わずかですが3000バイトほど空き容量が多くなります。また実行速度も割り込みが発生しない分だけ、いくらか上がります。

またIOCS.Xを組み込んでいる人もメモリが狭いと感じるようだったらはずしておくことをすすめておきます。スクロールの高速化などあれば便利ですが、なくても動くんだから我慢しましょう。また、FILESやBUFFERSの最初の数字も小さくすると多少はメモリ消費が抑えられます。ディスクアクセスが遅くなったり、同時に扱うファイル数に制限が出ますが「背に腹は代えられぬ」ってやつですね。

もちろん、このような操作も、ビジュアルシェルスで起動すると台なしです。真っ先にコマンドシェルスで起動するシステムディスクを作ってください。方法は各機種取扱説明書第3部「より高度な使い方」の3章「デスクトップを使わない操作」の4項「起動時にコマンドモードに入るには」を参照してください。



パソコンの画面をビデオに録ろうと思い、X68000のカラーイメージユニットを買ったのですが、市販のソフトウェアをビデオに録るときに、コンピュータの画面モードをスーパーインポーズすると黒が透けてテレビ番組が映ってしまいます。VCUTを実行しようとしても市販ソフトなので無理です。どうしようもありません。どうにかテレビ画面をカットする方法はないでしょうか。

静岡県 石井 孝



スーパーインポーズの状態でないでビデオ録画できないという制約がなければなんでもないことなのですが、どんなに考えてもスーパーインポーズさせないで録画できないのは仕様上、変更することは無理だと判断できます。

問題点はスーパーインポーズにあるのではなく、黒色が透明扱いされてテレビ番組が映ってしまうことなんです。

ということは、もしチャンネルをあわせるときに画面全体が真っ黒な放送があるとして、そこでスーパーインポーズしたらどうなるか。……そうですね、コンピュータ画面の黒(透明色)の部分にビデオ信号の黒が入って、うまくコンピュータ画面がそのまま録画できるわけです。

ところが、そんな放送があるわけがないので、どうやって黒色の画像を手に入れるかが問題となってきます。しかも、それを通してコンピュータ画面を見るのですから、ノイズの多いビデオ信号だと録画したときに画像が乱れて見にくいかもしれないので、できるだけ安定したものを探すことになります。

私の知っているものではセガマークIIIやメガドライブ、PCエンジンなどのゲーム機のカセットを入れずに電源を入れると、真っ黒の画面が流れたように記憶しています。ただし、これらは正確にはビデオで使っているビデオ信号とは微妙に異なる場合があるので、もしかしたら同期がずれたりノイズが出る可能性もあります。結局は手持ちのビデオ機器との相性次第ですので注意してください(録画側のビデオデッキにTBC機能がある場合はTBCをON/OFFして相性を調べてください)。

また、2台以上のビデオデッキがある場合、ほとんどのビデオデッキが外部入力にして画像を入力しなければ、画像出力側には真っ黒(灰色?)な映像信号が流れると思います。それらの出力をカラーイメージユニットのビデオ入力につなげておいてスーパーインポーズすれば、うまく録画できるでしょう。

なお、近日発売が予定されているビデオボード(カラーイメージユニットの録画専用版、イメージ取り込み機能はない)では内部にビデオ信号発生機を持っているのでこのような面倒な操作は必要なくなったようです。すでにカラーイメージユニットをお持ちなら特に必要ないと思いますが。



Oh!X1988年9月号のturbo RAY TRACERが動きません。リストを同封しますので、おかしところがあれば教えてください。

北海道 村松 良彦



村松さんの質問は便箋2枚にわたる長いものだったので、質問を簡略化させてもらいました。

ところで、送られてきたリストと質問の内容から判断すると、こちらの説明不足のため動作していない可能性もありますので、一応補足説明させていただきます。

記事ではリスト6からリスト9がデータの例として掲載されていますよね。これらのデータはリスト3のデータセットプログラムにマージして使うようになっているのですが、そのことが記事の中で触れられていません。たとえば、リスト6の例1が「EXAMPLE1」として保存してあるのなら、リスト3をロードしたあとに続けて、MERGE「EXAMPLE1」

のようにするのです。RUNすると、

INPUT FILE NAME:

と表示されますが、それにはリターンキーを押すだけで結構です。

これで駄目ならプログラムに入力ミスがあるものと思われます。(影山 裕昭)

質問にお答えします

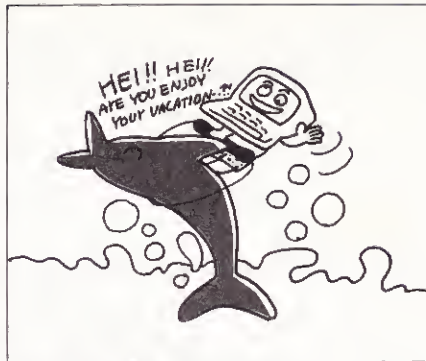
日ごろ疑問に思っていること、どんなことでも結構です。どんどんお便りください。難問、奇問、編集室が総力を上げてお答えいたします。ただし、お寄せいただいているものの中には、マニュアルを読めばすぐに回答が得られるようなものも多々あります。最低限、マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名、システム構成、必要なら図も入れてこと細かに書いてください。また、返信用切手同封の質問をよく受けますが、原則として、質問には本誌上でお答えすることになっていますのでご了承ください。なお、質問の内容について、直接問い合わせることもありますので、電話番号も明記してくださいね。

宛先：〒108 東京都港区高輪2-19-13

NS高輪ビル

ソフトバンク株式会社出版部

「Oh!X質問箱」係



FROM READERS TO THE EDITOR

もうすでに、外は夏真っ盛りかと思うほどの暑さ。でも、編集部の中はクーラーが効いているからすずしい、……はずな

のだが調子が悪いのか、なんか暑い。仕事なんかできな一。というわけにもいかず、しかたなく働くのであった。

◆本屋で「Oh!X」を手にしたとき、やっとディスク付録がついたかと思った。レジで「780円です」といわれて700円しか持っていなかった僕はスクーターをとばして80円を取りに戻った。ディスクを立ち上げて苦労しただけのことはあったと思った。

能美 和具(19)熊本県
すいませんねえ。しかし、予告で780円になりますと断ってあったとは思いますが。

5月号は立ち読みもしなかったのかな。
◆今月号は付録つきのためか異常に薄い! と思って購入したら「Oh!PC」ではなく「Oh!X」でした。貴誌の迫力(?)ある誌面に洗脳され我が家配備第1号はPC-286シリーズのハズがX68000PRO IIになりました。今後のご発展を祈ります(期待してます!)

坂本 慎太郎(29)東京都
やった、偶然とはいえ読者が増えた。しかし、Oh!PCのほうでひとり減ったのかな。同じ会社の雑誌だからなあ。

◆半年に1回ぐらいはディスクをつけてもバチは当たらないと思う。朝野 貴敦(17)滋賀県
いや、バチが当たって倒れる人が続出して本が出なくなるような気がします。

◆3週間に1度ぐらいディスクつきにしてほしい。
野田 佳照(16)愛知県
????????????

◆今回のディスクは僕の人生に常駐した。
中島 潤史(15)埼玉県
早くワクチンを使って頭の中を治療しないとボロボロの人生になってしまうかも(どういう意味なんだ)。

◆創刊8周年なんですね。ちなみに私のX68000PROは1歳です。だから、誕生日(買った日だよ)にはIMバイト増設してやりました。

山田 雅宏(18)岐阜県
このプレゼントで幸せになれるのはX68000か、あるいは本人か? たぶん両方ともにでしょうね。

◆6月は創刊8周年記念だということを知って自分の誕生日も6月なのでうれしかった。

佐藤 直人(11)神奈川県
実は編集部のE.O.さんも6月に誕生日を迎えたので、みんなでケーキを食べました。ひびきさのケーキはおいしかったなあ。

◆今回のようなディスク配布が不可能であれば、ダンプリストを圧縮した形で載せてもらいたい。入力が楽である。

塩谷 隆治(31)広島県
そうですか。じゃあ、来月からリストを1文字=0.1mmぐらいの大きさに圧縮して載せますので、皆さん虫眼鏡を用意して待っててください(冗談)。

◆いまYet Another Columnにはまっている。最初の頃は2時間ぐらい座りっぱなしだったが、いまでは自分の力量がわかってしまっただけで、2回ぐらいでやめてしまう。編集部には4万点をこすつわものがあるそうだが、どんな手を使っているのでしょうか。正当なやり方で取れるはずはないのだから、と思う。柴崎 誠(17)福島県
正当なやり方ですよー。どんなやり方かというところ……。

◆編集部での40,860点とゆーのは信じられない。おそらく、「必殺! ESCキーで止めればどこに落とせばいいかわかるじゃないか攻撃!」を使ったのだろーと思う。ちゃんければ、古代フェ



ニキアに生きた者の血を引ーてるとか。

大島 貴成(17)栃木県

実はですね、あのゲームには「なるべく高いところから落としたほうが点が高いんやでー攻撃!」というのがあるのですよ。気づいた人もいろいろいるようですが。

◆Yet Another Columnは面白かった。ブロックがくずれる音が「ケセランパサラン」と聞こえてしまうのは僕だけだろうか。そして、このゲームを「ケセランパサラン」と呼んでいる。友達におかしいといわれたが、なにがなんでも「ケセランパサラン」と呼んでいる。

奥村 真明(17)埼玉県

僕も自分のイスを「ギシギシくん」と呼んでかわいがっています。また、その友達には時計の「コチコチくん」というのがいます。

◆やられてしまった。かねてから作ろうと思っていた「この木なんの木」。日立社員のこの私が作ろうとしたのに会社が忙しくて……。P.S.習志野工場のお昼にはかかりません。さみしいなあ。ほかでは確かにかかるところもあるのに。ところで、一般人には「この木なんの木」が日立の社歌だと思っている人がいるようだが、これは社歌ではありません。社歌は別にあります。念のため。

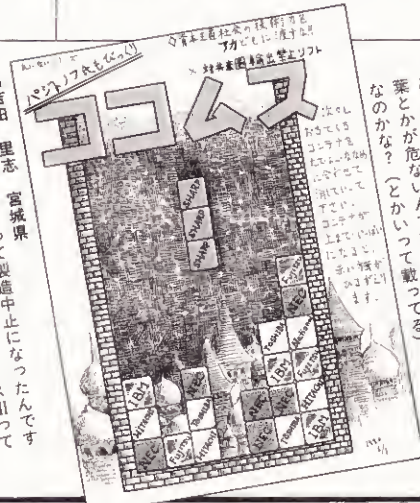
藤井 実(19)千葉県

アンケートハガキを読んでも「この木なんの木」がよかったというのが予想外に(?)多くてびっくりしました。別に「よかった」というのが予想外なのではなく数が本当に多くて。

◆Oh!Xを買いはじめて7,8カ月。そろそろ内容のペースにもついていけるようになりました。最初は内容についていけず(突然OPMAだとか書いてあったので)とても困り、なんて不親切な本なんだろうと思いましたが私もいいかげんなもので、いまではもっといろんなツール載せろーと心の中では思っています。これからもいろいろなプログラムを載せてください。次回の付録ディスクが近いうちにあることを願っております。では、さようなら。

小田 典央(19)静岡県

まあ、人間ってそういうもんですよ。



◆もっと本を厚くして5月号に載っていたX68000の変なデモみたいのをたくさん載せてもらいたい。

望月 伸幸(17)静岡県

変なデモというのはひょっとして(て)のショートプロバートに出ていた例のアレのことでしょうか。本が厚くなってああいふのばっかり載っていたらとてつもなく恐ろしいような気がします。

◆うーん、なにか押し入れの中でカサカサ音がするなあ……、と思って押し入れ開けてゴソゴソやっていたら、「ゲッ!」、思っていたとおりゴ、キ、ブ、リとご対面してしまった。予想していたこととはいえ、やっぱり気持ち悪い。と、躊躇していたらフトンの中に逃げ込まれてしまった。おそらく、まだ中にいると思われます。田舎にいた頃は東京近辺よりは湿度が少なかったせいか、ほとんどゴキブリは見たことがなかったのですが……。ゴキブリを見かけたせいで、「ああ、俺も関東に住んでいるんだな」と妙に感激してしまった。が、やっぱりイヤだなー。でも、早めにやっつけんといかん。うん。

工藤 隆(20)埼玉県

ゴキブリのもっとも恐ろしいところは……。それはやはり、叩き殺そうとしたら顔に向かってバタバタと飛んでくるところでしょう。あの瞬間のこわさきたらこの世で1番じゃあないかと思ったりします。

◆僕の友達が考えた“パソコンとカツ丼を手に入れる方法”。

- 1) 展示パソコンを持って逃げる
- 2) 逃げる途中に隠れている友達にパソコンを渡す
- 3) わざと警察に捕まる
- 4) 黙秘権を使う
- 5) しばらくすると警察がカツ丼をくれる
- 6) もうちょっとすると釈放される

これでパソコンが手に入りカツ丼も食える。

すごい! 小川 伸一郎(15)京都府
いやー、すごいですね。15歳(?)にしてこの頭脳。編集部一同思わず感心してしまいました。まさに完全犯罪ですね。ひょっとしてノーベル賞ももらえるかも。どうもおめでとうございます。

◆X68000も10万台をこえたようなので、そろそろマニア以外にも売れることを考えたらどうだろう。案としては自己診断機能の高度化。たとえば、まずコンセントを入ると周辺機器をチェックする。ディスプレイやキーボード、マウスが接続されていないと、「私の顔をつけて」とか「私のねずみはどこ」と話して誰でも接続できるようにする。笠井 康彦(23)神奈川県

すると、接続を間違えたりすると「そこじゃないわよ」とか、スイッチを切ろうとすると「やめて」とかしゃべるんだらうか。あー、気持ち悪い。

◆PC-9801と同じくらい普及しているビジネスパソコンであるといつてX68000を買ってもらったのに例のウイルス事件によってうそがばれ



◆杉本 秀昭 宮城県
なんかよくわからないけど、とてもと風邪をひきますよー。とかなくそんな格好らんこといつたりして。とかなんと、いい



◆大村 直人 北海道
どうもありがとうございます。やはり、お祝いのハガキもないね。しかし、こういうわいのある(?)イラストですね。しかも、とっても味

た。森下 剛(14)京都府
そんなすぐにばれるようなうそを……。

◆最近、アクションゲームやロールプレイングゲームに興味がわなくなった。どうしてだろう。中井 卓(18)大阪府

どうしてだろう。きっと大人になったんだ

◆涙の浪人生活に入ってから小遣いを1,000円に減らされてしまった。しょうがないので弁当を作ってもらえなかった日に食事をぬいて300円ほどひねりだし、やっとOh!Xと好きなバイクの雑誌を買っている有り様。なんとも情けないことであります。しかたないですけどね。最近ではゲームもあんまりしてなかった(というより、「これ!」と思うのがなかった)ポピュラスを知ったとき、はまってしまいそうでこわいと思ひながら金がないのでさみしく思っていました。そこにこのプレゼント。僕にポピュラスをくれー。Oh!Xを買い始めて7年目。小学生だった僕もいまは浪人生、なんかすごいものを感じるなあー。安陪 亘(18)三重県

ううっ、なんて情けない。ごはんを抜いてその浮いたお金でなにかを買うというのはよくある話ですが、体をこわさない程度にしましょう。でも、そうかといってポピュラスをあげるわけにはいかない。

◆ふと思った。ファジィコンピュータ内蔵(ぢゃなくて内蔵)のカメラで撮った写真はどのようになるのか。

fuzzy(形)「中略」2. [写真が]ぼやけた(blurred)ーシニア英和辞典 4訂版よりー

大村 直人(17)北海道
なるほど。

◆暑さが厳しくなっているなか、部屋に閉じこもりっぱなしだと頭がどうにかなりそうです。懸賞にクーラーもつけてください。

荻久保 雅道(14)静岡県

僕もクーラー欲しい。

◆以前、続けて4回足を運んだ映画のサントラ盤をステレオを持っていないのに買ってしまっ

三原 克之(36)福岡県
そういえば、僕もCDラジカセしか持ってないのにレーザーディスクのソフトやレコードを持っている。

◆X68000が10万台前後だそうです。もし個人でソフトハウスを開業したとして1パーセント以上の人が(通信販売で)ソフトを購入すれば経営が成り立つと思います。「私はやってみたい!」と思っている人はかなりいるのではないのでしょうか。ですから、ソフトハウス経営についての特集をお願いします。特に、ダビング工場のメーカー名と連絡先やその手数料、パッケージの単価と依頼数量など。この特集をすることにより、X68000ユーザーの中からソフトハウスを開業する人が多く出る→ソフトが増える→X68000購入者が増える→Oh!X購入者が増える! 高久 裕明(29)東京都

やはり、問題はその個人が作ったソフトが市販ソフトとして受け入れられるようなレベルに達しているかどうかでしょう。つまらなければ、やっぱり全然売れないだろうし、面白ければ販売しようという話はどこから来るでしょうから。

◆ANGELの人体モデルはどうして女の人なので

すか。竹永 昌伸(16)兵庫県
うっ、それだけは聞かないで。じゃなくて、ただ単に男だと気持ち悪いからじゃないでしょうか。

◆アンケートハガキの何パーセントが読まれているのだろうか。読まれなければなにを書いても出さないのと同じだもん……。小杉 雅信(21)愛知県

全部読んでに決まっているじゃないですか。このコーナーやハミダシっていうのはアンケートハガキによって成り立っているんですから。だから、白紙とかでなくなんか面白いことを書いて出してください。スタッフの人なんかにもくたびにハガキを読んでますよ。

◆いつもOh!Xの記事を見て、すごくらやましくなります。なぜかといえば、SHIFT BREAKとかmicroOdysseyとかみたいに自分の考えを自由に(多少は制限があるでしょうが)書いて、ま

たそれに対して読者から意見がきて、またそれに対して意見を言えるという。なんか、そういうのっていいですね。いちばんうらやましいのはやっぱり「STUDIO X」の答える人かな。一度でいいから代わってほしいと思うのは僕ぐらいなものでしょうか。 斉藤 哲哉(18)愛知県
そんなにうらやましいですか？ まあ、一応仕事としてやっているんですが、確かに自由に書いたり、その反応が返ってくるというのは実に楽しいことです。

◆気がついたら、知らない人の家にいた。大学の芝生の上に寝ていた。梅田の映画館の中にいた。先輩、日本酒とビールのカクテルの中に味の素、塩、魚の頭、キャベツ、しょう油を入れて飲ませないでほしいな（文科系サークルとは思えないところに入った……）。

佐藤 能久(19)大阪府
いや、体育系より文科系のほうが飲み会がきついというのはよくある話ですね。しかし、魚の頭やキャベツだったらいいですよ。もっと、ひどい話を聞いたことがあります。それは、……（あまりにもひどくていえない）。

◆X68000のスーパーインポーズでうそのニュース速報（チャイムつき）を流し、バアさんを指名手配の犯人に仕立て上げたら、バアさん3日間悩んだ。 松本 浩一(24)栃木県

僕もそういうことを考えてPC-6601SRでやろうと思ったのですが、グラフィックが粗いので漢字がでかくなるし、第一、専用ディスプレイがなくてスーパーインポーズができなかったのです。ううっ、悲しい思い出だな。

◆ゆるせないぜ！ アンケートハガキの下の“X68000（無印、ACE、PRO……）”の無印てのはなんだよー。初期型はなー、グラディウスが付いてたんだぞ。CZ-6000万歳！

御宿 桂治(18)山梨県
何をいってるんです。無印良品っていうじゃないですか。うーん、しょーもない答えになってしまった。

◆いま気がついたのですが、アンケートハガキの裏表に年齢を書く場所があるのには意味があ

るのだろうか（すでにどなたかが気づいているかもしれないが）。もしかして、裏の年齢は愛機の年齢を書くのだろうか。

西谷 健吾(17)兵庫県

違います。裏には数年を書くんです（またまた、しょうもない答え）。

◆HDタイプのX68000は地震に弱いので対策を立てました。それはキャリングハンドルを利用して天井からロープで吊るすのです。そうすれば、ソバ屋の出前バイクの法則によりX68000は地球の重心に対して静止するのでクラッシュの魔の手から逃れることができます。ぜひ、おためしください。それにしても大洋は強い。

矢地 雄(18)東京都

部屋が広ければ問題はないけど、せまかったらロープの長さによっては悲惨なことになるそう。壁にぶつかって。そうでなくても、落ちたときのことを考えると、とてもおためしなんかできない。

◆なんというとか。「ハード」のプレゼントがないじゃないか！ 私は楽しみにしていたのに（当たるわけもないけど……）。今月号はX68000を持っていればとってもうれしいのかもしれないが、ほかのユーザーはどうしろっていうんだ。

秋友 謙二(16)山口県

「ハード」のプレゼントは今月だったんですよ。はっはっは。しかし、なかなか当たるのは難しいでしょうね。

◆読者の方に聞きたいんですけどマウス、トラックボール、みんなはどっちを使っているのでしょうか。私の場合、部屋が狭い（4畳半、バス、トイレ、キッチン共同で家賃8,000円。今春から1,500円上がった。くるしー）ので机の上にキーボードとサイバースティックを置くといっぱいになり、マウスとして使うスペースがなくてトラックボールとして使っています。両手はふさがりますが、そのぶんマウスのときのような腕の筋肉痛（あるわけねー）がなくなります（運動量が少ない）。みなさんはどっちです。

栗 幸司(21)広島県

僕はマウスとして使っていますが、机の上の空きスペースが10×10cmぐらいしかないので非常に苦しい。

◆初のフロッピーディスクの付録、年寄りには最高のオマケでした。長いリストを打ち込むことは体力が持ちません。最近ではリストを見るだけであきらめていたものでした。年寄りのためにもこれからときどき入れてほしいと思います。

小池 清(42)滋賀県

年寄りというほどの年でもないと思うんですが。まあ、長いリストを打ち込むのってけっこう体力が必要ですね。

◆愛読者年間モニタの応募者が欠員というのは、とても残念です。読者の皆さんがどうせなれないだろうと敬遠しているのか、本当に参加意識が薄れているのかはわかりませんが、7名というのには驚きです。私は第1期のモニタをさせていただいたので前者のほうですが、モニタ経験のある者でももう一度できるものならぜひやりたいところです。たぶんあの記事に刺激されていまではかなりの数の応募があると思います。

紺谷 憲児(22)大阪府

別に一度やったからといって、年間モニタが二度とできないということはありませんから、経験者の方もどんどん応募してきてください。

◆1年ぶりにX1turboと再会した。が、2,3回スペースキーを叩くとスペースキーが死んだ。こうなるとほとんどのゲームができない。しょうがないのでワープロとして無理に使っていた。でもこれでは面白くないので、近くの電器屋に修理に出したらキーのスイッチとカールコードの交換で1万円以上もした。おかげで翌日のビジネスショウに行けなくなった。しかもである。スーパー大戦略をやっている気がついたのだが、HELPキーが死んでいる。どーしよう。あんまり使うキーでないだけに悩んでしまう。

加藤 健二(18)埼玉県

まさに「一難去って、また一難」。

◆やっぱりX68000はいいですね。あつ、そういえば4月のいくんちだったか忘れてましたが、夜、MOTOSを立ち上げたらいつものオープニングの曲と違う曲が流れたんです。あれは、なんだったんでしょう。 野口 智広(17)神奈川県
さあ、なんだったんでしょう。

◆バットモービル届きました。こんな凄いプレゼント生まれて初めてです(笑)。とりあえずディスプレイの上に飾ってあります。暇になると走らせてみたりしていますが、傍から見るとちょっとあぶないやつに見えるかも（かもじゃないって）。 松久 孝治(20)岐阜県

走らせるときに「ブーン、ブーン」とかいうてやると、なかなかいいかもしれない（ながいのがいいのやら……）。

◆はじめまして。僕はX68000を買って（もらって）1年と少したちました。買って1ヵ月ほどたってから今まで、BASICを興味だけで学んできました。自分ではなかなか進歩したと思って、そろそろ高レベルの雑誌を購入しようと思いOh!Xを買うにいたったのです。が！ 内容を見たとなん、全身の血が凍ったかと思うほどにおどろいた（なんじゃそりゃ）。今まで僕がコッ

Super Intelligent Book Series.

Oh!X 別冊

清涼飲料水

全カタログ

あなたはここに掲載されている全ドリンクを制覇出来るか。

古村 聡 著

SOFT BANK

清水 健年 東京都
こんな本が出たら売れるんだらうか。うーん、やっぱり結構売れてしまうんだらうな。しかし、読んだら飲んでみたくなるんだらうな。

MAKE YOUR LANDS POPULOUS

信川 洋 東京都
ポピュラスって本当に人気が高いんですね。シナリオディストロミストランドも出たことだし、ますますハマル人が続出するのかな。

ツ学んできたことは、まるで宇宙の中の人間……まではいかないが、星のような(あんまし変わらん)ものだったのです。そんで、今月号の中に「PROを買って半年たって……」などといったが僕にはわけのわからんことが書いてあったりします。いったいこーゆー人はどうやって学んだのか……。教えて！

田村 高志(16)愛知県人
人は人、自分は自分ですから、マイペースでコツコツやるのがいいんじゃないですか。試行錯誤しながら自分で学んだほうが身につくし、面白いですからね。

あとがき

ここに載っているのは6月号のハガキからなんですが、6月号のアンケートハガキはやはり、ディスクに関するものが多かったですね。目についたものをちよっとまとめてみました。

・Yet Another Columnが面白くてはまった(定期試験があるにも関わらずとか、忙しいのにとかいうのが多い)。

ちなみにハガキで書いてきた中のハイスコアは滋賀県にお住まいの小野さんの50,875点で

した。

・ディスクつきになるんなら、〇〇円出しても買う。

・ディスクをフォーマットしてしまった。

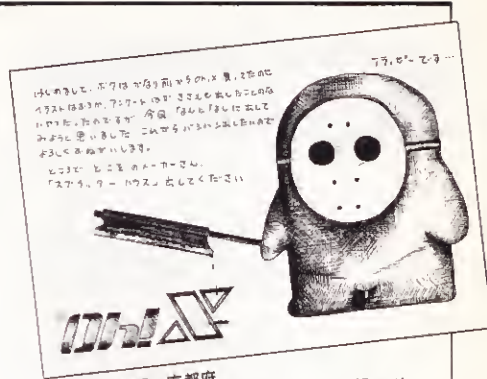
こういう人が何人かいましたが、やはりプロテクトシールを貼っておいたほうがよかったのかな……。まあ、そういう場合はしょうがないのでそのディスクをこちらへ送ってもらえればもう1回書き込んで送り返してあげられると思います。

・6月号は保存版にしたいので2冊買った。

・ANGELが走らない。

ANGELはまったく走らないと思っている人が多いようですが、そんなことはありません。

勘違いをしている人がいるかもしれませんが、6月号の49ページの「2) FLOAT2+, Xの動作がおかしくなることがある(たまたま計算を間違う)。ANGELは実行しないように。」というのはFLOAT2+, Xを使ってANGELを実行しないようにという意味で、ANGEL自体がおかしいということではありません。FLOAT2, Xを使えばANGELはちゃんと動きます(ANGELに



▲熊谷 逸郎 京都府
なんでフロッピーがスプラッターハウスに……。なんでか似合ってますよね。マスクを取ったときに爆笑されそうだけど。

は多少おかしいところもありますが、起動は絶対にはずです。走らない人はもう一度本文をよく読んでください。

なんか違うコーナーのようになってしまいましたが、今月は大部分がディスクに関するハガキでしたので、あしからず。

ぼくらの掲示板

仲間

★「バ〇レイバー」のサークル「狂走都市」ではCGを大々的に扱っている会報を発行中です。CGのPDS発行も企画しています。「バ〇レイバー」だけでなくほかのゆうき先生の作品をCGで描いてみたい15歳以上の方、入会してください(当会所有機種はX68000 EXPERT, PC-9801RX21)。62円切手同封でご連絡ください。〒949-66 新潟県南魚沼郡六日町大字六日町484 種村聡人(15)

★X68000ユーザーを対象とした(X1でも可)サークル「X68K-GR」の会員を募集しています。ゲームを中心に月1回程度の会合を開き、ゲームの情報交換や会誌の発行をしたいと思ひます。興味のある方は簡単な自己PRと62円切手同封のうえ、下記の住所までご連絡を。〒239 神奈川県横浜須賀台ハイランド5-3-19 三浦正義(17)

★「ディスクサービスX68000」。このサークルではX68000の機能をより高度に活用するためにユーザーの方を募集しております。活動内容は皆さんから集めたプログラムを会報にまとめ、配布するというものです。興味のある方は62円切手同封のうえ、ご連絡ください。〒165 東京都中野区上鷲宮5-28-24 前野千絵(21)

★このたびサークル「べけろく亭」では会員増強に伴い第3期会員を限定募集いたします。パソコンをより高度に活用するための情報交換などが主体です。入会希望の方は62円切手同封のうえ、下記までご連絡ください。〒491-03 愛知

県一宮市萩原町富田方字茶原54 野村真広(23)

★このたび、X68000ユーザーのサークルを作るにあたりましてプログラマを募集しようと思ひます。できればアセンブラやC言語のできる方、興味があつてこんなボクでも力になれるなら……とか思つた人はぜひご連絡を。〒437-11 静岡県磐田郡浅羽町浅羽1169-32 袴田信孝(17)

売ります

★MZ-2000用周辺機器を以下の価格で(送料込み、値引き可)。フロッピーディスクドライブ「MZ-1F07」を2万5千円、プリンタ「MZ-80P6」を1万円。いずれもインタフェイス、ケーブル、説明書つき。拡張ユニット(MZ-1U01)、漢字ROMボード(MZ-1R13同等品)、VRAM3ページを各8千円で。連絡はハガキで。〒371 群馬県前橋市上小出町6-1-205 中嶋康弘(31)

★X1用データレコーダ「CZ-8RL1」を5千円(送料込み)で。連絡は往復ハガキで。〒737 広島県呉市阿賀中央1-24-20 清水幹雄(18)

★MIDI音源ローランド「D-110」(1年使用)をマニュアル、保証書、付属品一式、1Uラックつきで4万5千円で譲ります。送料当方負担。連絡は往復ハガキで。気長に待ちます。〒594 大阪府和泉市鶴山台4-8-3 山路智弘(20)

★IO-735X(カラーイメージジェット)、マニュアル、付属品、X68000用ケーブルつき、箱あり、キズなしを送料込み8万円で。連絡は往復ハガキで。〒238 神奈川県横浜須賀台深田台76 小林秀樹(21)

●掲載ご希望の方は、官製ハガキに項目(売る・買う・氏名・年齢・連絡方法……)を明記してお申し込みください。

●ソフトの売買、交換については、いっさい掲載できません。

●取り引きについては当編集部では責任を負いかねます。

●応募者多数の場合、掲載できない場合もあります。

★ワープロ書院「WD-540」を5万円で。箱、マニュアル、付属品、保証書あり。おまけでインクリボン、フロッピーディスクをつけます。連絡は往復ハガキで。〒187 東京都小平市花小金井4-286 小川和幸(24)

★増設RAMボード「CZ-6BE1」を送料込みで2万2千円程度で。少しぐらいなら割引します。マニュアル、付属品あり、キズなし。連絡はハガキか手紙でお願いします。〒665 兵庫県宝塚市南ひばりが丘3-26-5 関口敬文(15)

買います

★X1用FM音源ボード「CZ-8BS1」(完動、箱、マニュアル、付属品あり)を送料込みで8千円から1万円で。連絡は往復ハガキで。〒673 兵庫県明石市西明石北町3-16-5 春名隆行(17)

★MZ-2521用辞書ROM、増設RAMボード、増設VRAMボードを各6千円ぐらいで。コンパチ品も可です。希望価格を明記してハガキで。〒458 愛知県名古屋緑区鳴海町神明163-1 安川実(16)

★X68000用増設RAM「CZ-BE1」を1万5千円ぐらいでお願いします。完動で付属品つき。連絡はハガキで。〒285 千葉県佐倉市井野869-26 松本琢磨(17)

バックナンバー

★Oh!X1989年5月号を千円、「X68000テクニカルデータブック」を2千円で。美品希望。連絡はハガキで。〒289-13 千葉県山武郡成東町成東2470 安井忍(22)

▶最近このコーナーで紹介しているサークルに対して、問い合わせでも返事がこない、他人の著作権を侵害する行為が行われているなどの残念な報告が増えています。信頼関係に基づくコーナーですからマナーの向上に留意してもらいたいです。今後、仲間のコーナーでの紹介を希望されるサークルは必ず会誌の見本を送ってください。

DRIVE ON

このコーナーでは、本誌年間モニタの方々のご意見を紹介しています。今回は6月号の記事に関するレポートです。これまでのモニタの方にとっては最後のレポートです。1年間のモニタレポートご苦労さまでした。

●「共通システム」という考えに基づいたS-OS思想は、5年たったいまでも決して古びてしまったものではない（それどころか、いまだからこそ重要なことかもしれない）。しかし、「SWORD」というシステム自体は8ビット全盛時代のものでしかない。16ビット以上の時代の8ビットのためのシステム、「Excalibur」なり「Storm Bringer」なりを発表すべきではないだろうか。INTEGRAL XI (KAME-DOS)のように、どんなディスクフォーマットも読める機能やヒストリといった近代的機能を備えるS-OSが発表されてもよいと思う。だからといって、テーブルユーザーや旧機種ユーザーを切ってもいいというわけではないが。西田宗千佳(18) X68000, X1Fmodel20 千葉県

●なんといっても「PurePASCAL」、これに尽きるのではないのでしょうか。グラフィック機能が標準でないのが残念ですが、これが付けばかなりなものになると思います。私のようなPASCAL派はX68000ユーザーの中には少ないかもしれませんが、これを機にPASCALを勉強してPASCALの素晴らしさを知ってほしいと思います。できれば、コンパイラの内部仕様などに関する記事があってもよかったのではないかと思います（まあ、これは今後に期待しましょう）。

森川一(24) X68000ACE-HD, XlturboII 北海道

●「ハードウェア工作入門」についてですが、製作の対象とする回路はできるだけシンプルなものをお願いしたいと思います。また、なるべくローコストでということも。最近では気軽にハンダゴテを握ったり、紙工作したり、プラモデルを作ったりというような話をあまり聞かなくなりました。これはやはり、身の周りに完成品があふれているためでしょう。でも、誰でもみんなにかしらの創作意欲を持っているはず。「ハードウェア工作入門」には、そんな私たちの創作意欲を刺激し満足させてくれる連載になってほしいと願います。

藤田康一(19) X68000PRO 静岡県

●S-OSがまさかX68000やPC-286にまで広がると思っていませんでした。うれしいかぎりです。でも、X68000ユーザーはともかくとして、PCユーザーがこのことを知らないのは残念だと思います。Z80シミュレータとしてなんとかPCユーザーに知らせる方法はないでしょうか。ちなみにPC-9801RSで動かしてみましたけどXIよりも少々速いような気がしました。なんといっても2Dのディスクの読み書きができるのは5重丸です。

末吉克行(21) X1G, MZ-731, FM-7 兵庫県

●「ハードウェア工作入門」ですが、前回のアンケートで書いたことはちゃんと押さえてあり、「何が必要であるか」ということがわかりやすく書いてありました（さすがだなあ）。プログラムのようには、簡単にはやり直しが効かないハードウェアが相手ですから、なかなか大変だと思います。入門講座の場合いちばん大切なのは、「急に難しくならない」とことだと思います。余談になりますが、NHK基礎英語がいまだに入門講座として利用される

ことが多いというのは「急に難しくなることがない」からなのだそうです。そうしない、ついでこれないというわけです。バカ丁寧すぎるくらいでいいですから、ゆっくりのんびりやってほしいですね。それと、なるだけわかりやすい図を使ってほしいと思います。ジョイスティックポートにつなぐものがほとんどと聞いて安心しました。実際に組み立てる場合、回路図と配置図がバツとは結び付かないものです。毎回言っていることなのですが、「難しいことばは、脚注などを付けてもらいたい」と思います。X68000マシン語講座がなぜ読みやすいかというと、脚注などが詳しく、難易度に気を配っているからだと思います。ハードとソフトの違いはあるとはいえ、やっぱりこうあってほしいと思います。湯澤聡(27) X68000, XlturboIII, MZ-286/2531, PC-6601, MSX, PC-1360K 埼玉県

●アンケート結果を見て。やってくれますねー。まあ、そうとうの内輪ネタであるのですけれど、こういった内容であればいたしかたないでしょう。さすがのX68000の伸びと、ほかの項目内のX68000が占める割合が、いまいちばん私にとってショックですね。あのとき、XlturboZかX68000が多少なりとも悩んだんですから。ベストライター（もちろんOh!Xのスタッフ1人ひとりにはベストライターです）、なんていうアンケートはまさか載せるためだとは思っていませんでした。祝一平氏がトップなのは、やはりという感じ。ま、ほかにもいろいろありましたが世論調査みたいでいいですね。作り手と受け手がこうもコミュニケーションできるのはOh!Xだけでしょ。また、やりましょう。大津和之(20) XlturboZ 福岡県

ごめんなさいのコーナー

7月号 AFTER REVIEW

「サーク」のレビュー内の写真が「ルーンワース 黒衣の貴公子」のものに入れ替わっていました。関係各位にはご迷惑をかけました。お詫びいたします。

7月号 WZD

先月号のものではコマンドラインからパラメータ付きで実行した場合、復帰時の動作が保証できません。詳しくは今月号のP.147をご覧ください。アセンブル時は問題なく動作するはずですが。

6月号 ANGEL

P.65 回転のコマンドの書式に間違いがありました。

rotx <式> → rot.x (<式>)

roty <式> → rot.y (<式>)

rotz <式> → rot.z (<式>)

のように変更してください。お詫びして訂正いたします。

また、画面をはみだすような絵を描かせることと止まってしまうことがあるようです。

6月号 GCC Ver.1.36.01

Humanのバージョンが2.00の人はいま動くかないようです。前にも書いたようにHuman v.2.00はシャープでv.2.01に交換してくれまので、これを機にバージョンアップしましょう。

6月号 X68000マシン語プログラミング

files.hのリストが抜けていました。詳しくは今月のX68000マシン語プログラミングをご覧ください。申し訳ありませんでした。

バグに関するお問い合わせは
☎03(5488)1311(直通)
月～金曜日 16:00～18:00

お問い合わせは原則として、本誌のバグ情報のみに限らせていただきます。入力法、操作法などはマニュアルをよくお読みください。また、よくアドベンチャーゲームの解答を求めるお電話をいただきますが、本誌ではいっさいお答えできません。ご了承ください。

本誌創刊号 をプレゼント！ そんなバカな〜っ

▼お陰さまで本誌は通巻100号を迎えることができました。今月号は、グラフィック特集、表紙ぎやらいなどでカラーページを増ページして豪華にお送りいたしましたがいかがでしたでしょうか。

さて、編集部ではこれを機に取り置ききのバックナンバーを整理し、その一部をなんらかのかたちで皆さんに提供したいと考えています。とりあえず今回は100号記念プレゼント番外編として、Oh!MZの創刊号を3名の方に差し上げたいと思います。ご希望の方は縦じ込みのアンケートハガキのプレゼントNo.に0と記入してお送りください。

▼本誌では、コンピュータサークルなどの制作による同人ソフトの紹介を考えています。特にX68000などのユーザーグループの作品にはレベルの高いものが多く、市販ソフトにはない手作りの味が魅力です。これらはパソ

ケットなどを通じて安価に販売されていますが、一般にはあまり流通していません。本誌ではこうした作品を広く読者の皆さんに知ってもらいたいと思います。本誌での紹介を希望するソフトがありましたら、編集部までご連絡ください（☎03-5488-1309）。また、団体名、連絡先、代表者名を明記のうえサンプルソフトをお送りいただければ幸いです。

▼ここで嬉しいお知らせです。しばらく本誌を離れていた清水和人氏が次号より復帰。ゲームやプログラミングの楽しい記事をお願いすることになりました。ご期待ください。

▼先月号でお知らせしたとおり、7月1日から株式会社日本ソフトバンクは「ソフトバンク株式会社」と社名を変更しております。また、社屋も移転となり、Oh!X編集部は16日より新しい編集部にて業務を開始しております。お問い合わせの際には、電話番号が変わっておりますのでご注意ください。

▼先月号に掲載した日コン連企画協会の広告中、XIユーザーに対して不適切な表現があり、ご迷惑をおかけしました。広告主になり代わり、深くお詫び申し上げます。

投稿応募要領

- 原稿には、住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。
- プログラムを投稿される方は、詳しい内容の説明、利用法、できればフローチャート、変数表、メモリマップ（マシン語の場合）に、参考文献を明記し、プログラムをセーブしたテープ（ディスク）を添えてお送りください。また、掲載にあたっては、編集上の都合により加筆修正させていただくことがありますのでご了承ください。
- ハードの製作などを投稿される方は、詳しい内容の説明のほかに回路図、部品表、できれば実体記線図も添えてください。編集室で検討のうえ、製作したハードが必要な場合はご連絡いたします。
- 投稿者のモラルとして、他誌との二重投稿、他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒108 東京都港区高輪2-19-13 NS高輪ビル
ソフトバンク出版部
Oh!X「㊤㊶㊷」係

S H I F T ・ B R E A K

▶マーク・トウェインの「不思議な少年」を読んだ。天使「サタン」が人間の矮小さ卑俗さを描いてみせる。そのなかに彼が泥の小人を箱庭に生活させ、城を作ったところで雷で皆殺しにするくだりがあった。彼は泥人間の運命なんか気にもしていない。なんて残酷な奴だ。……さて、読書はここまでにしてポピュラスでもやろーかな。今日は272面だ。（H.U.）

▶5月号でレビューした天下統一だが、実はとても速かったのだ！ 愚かにも「もうCPUの速さがでちゃうんだよう」などと書いたが、なんのなんの、製品版は80286のRXにも決してひけをとらない。いや、それ以上だといえる！ こんなところでフォローしても何人の人が読むかわからないが、とにかく！ おもしろい！（電）

▶出張でスペインとポルトガルに行くことになったのだが、出発予定日の3日前になっても飛行機が何時に成田を出るのか知らされていない。それどころか、旅行会社は昼出発と言い、航空会社は夜出発だと言い張る。おまけに、旅行会社から聞いたポルトガルのホテルの住所は実在しない地名だと言う噂だ。果たして無事だどり着くのだろうか。（K.M.）

▶ポピュラス全500面クリア達成しました！ でも、エンディングのようなものではなく、0面（GENESISのようなもの）に戻ってしまうんですね。もしかして、2周しないとエンディングが見れないとか!?（R-TYPEみただな）今度は「対戦」と「プロミストランド」を究めてみようかと思ひます。Uさんこの間の勝負は練習ですよ、フフフ。（善）

▶ちょっとお尋ねしますが、皆さんの中でトマトジュースのお好きな方はいらっしゃいます？ 僕はあれが好物なのです（野菜ジュースはもっと好き）が、友人から毒物飲料のごとくいわれてしまいました。健康飲料と称した妙なのがいっぱい出てくるずっと前からあった、由緒正しい飲み物のはずなのに（関係ないか）。誰かご賛同を！（A.T.）

▶以前自分が担当してたころの質問箱のページを読み返してみたら、上手く書けてんだよ、これが。内容、文章ともに完璧に近い。俺って凄かったんだなーって本気で思ったね。つまりいたり、水たまりに落ちたり、大の〇〇踏んづけたりするのを気にせず空を見上げて歩いているほうが気持ちいい。ということらしい（おや、雨だ）。（Mu）

▶なれど、高校生の頃から大嫌いだっただが「自然保護」という言葉であった。だって、保護というのは「自分より弱いものを守ること」ではないか。いつから人間は自然を保護できるほど偉くなったんだ？ いつからそんなに傲慢になったんだ？ もっと謙虚になりなさい。謙虚に。そして、正直に「人間保護」とでもいってなさい。私は悲しい。（K）

▶ネット上のジョークを真に受けるヤツ。7月を待たずにウィルス終結宣言をするお役所。しかし、ウィルス学会というのはウィルスを作ろうという学会だったとは……。さて、POPULASの決め手は序盤。2つ目の城を何秒で作るか、いかに海を制し、どれだけ速く侵略できるかにかかっている、と思う。マップのせいにしてはいけませんよ。（S.N.）

▶新社屋となるNS（日本ソフトバンクではなく、日本食堂の略）高輪ビルへ見学に。下には富士銀行、上にはレストランという結構な趣。ところでその日は変な考えばかり浮かぶ日で、社長室を見ては「6万円ぐらいで貸してくれないかな」とか、帰りにNECのスーパータワー（風穴のあいたビル）を見て「あつ、クレイジークレイマー」とか……。 （A）

▶自慢じゃないがシリーズその2。私はいわゆる霊現象によくあう。台所に鑑測者が出たり、遠体離脱や寝入り端の子守歌なんてのはザラ。予知夢も多いし、デジャヴってやつも日に3回くらいある。ほら、こうやって原稿を書いているのだったってあった気が……。そういうや昨日にも、その前にも……。ん？ そりゃ単なる習慣だって。（E.O.）

▶創刊100号。私が編集に加わってから52冊、半分以上になるのか。半年前まで最若手だったのに……。編集部が大使館立ち並ぶ千代田区からお寺の並ぶ泉岳寺へ移転することになった。思えばここも3年半。さらば、武道館、靖国神社、北の丸公園……テキ屋にダフ屋の群れ、50mごとに並んだ警官の列……。さらば白百合学園のセーラー服。（U）

▶おかげさまで6月号は売り切れ店続出、なかには500冊以上売っていただいた書店もある。1年間バックナンバーが買えるよう在庫を増やしたのだから……。さて、その6月号にゲーム基板の話があったが、文脈上Oh!FMのY氏が基板評価に関わっていると誤解を招く部分があり、Y氏には申し訳ないことをした。この場を借りてお詫びしたい。（T）

microOdyssey

東京オリンピックで日本がアルゼンチンに勝ったときは、まだ私にはサッカーのなんたるかがわからなかった。目覚めは2年後のワールドカップ・イングランド大会の決勝で、地元イングランドと西ドイツが同点で延長戦に入り、結局イングランドが4-2で勝ったときだ。勝ち越しの1点は、バーに当たって落下し、ボールは外に跳ねかえったが判定はゴールであった。

抗議する西ドイツ選手たちを静めたのはキャプテンのウヴェ・ゼーラー。次のメキシコ大会の準々決勝で再びイングランドにリードされたが、なんとロスタイムにゼーラーはゴールに背を向けたままヘディングシュートを決めたのだ。準決勝はさらに激しいイタリア戦。肩を脱臼したベッケンバウアーがギブスで腕を固定してプレーを続ける姿は子供心に焼きついている。西ドイツはやはり終了間際に同点、延長で逆転。が、再度逆転され、さらに追いつくという歴史に残る死闘の末に敗れた。以来、私はずっと西ドイツの熱狂的(?)ファンを自称している。

ああ、それにひきかえ、なんでこんなに弱いんだろうと悲しくなるのが日本のサッカーだ。日本よりも弱い国なんて世界中さがしてもそんなにはない。それも競技人口からいえば結構大國に属するわけで、「いや日本じゃあまり盛んじゃないから……」と言いつつもきかないのだ。

日本が弱い理由は、1) 技術がない。2) 体力がない。3) センスがない。の3点が基本だが、もっと深い部分、思想的な面で問題があるような気がする。サッカーだけでなく。チームプレーを必要とする球技は基本的にダメなのだ。

たとえば、子供たちのサッカーで、キープ力のある子がドリブルで突破しようとする、その子だけが「1人でやっちゃダメでしょう」と注意を受ける。周りの子供が注意されることは意外と少ないものだ。チームプレーに「力を含ませて、助けあい」というイメージが植えつけられるのはこのときからではないか。

プロのサッカー選手が誰かにパスを出すのは、それが自分にとって「もっともいいプレー」となる場合だ。逆にボールを持たない選手はパスをもらえる状況を作るのが仕事の基本だ。

ボールを持つ選手Aはできれば自力で突破したいし、その自信もある。だが別の選手Bが、いやオレにつなぐのがお前にとってのベストチョイスだといわんばかりに動く。Aは、しかたがない、いったん任せるがリターンをよこしたほうが身のためだぞ、と前に進む。この駆け引きの結果が真のチームプレーとなる。パスは助け合いではなく仕事なのだ。

また、精神面でも多くの教育的指導は勝負に向いていない。たとえば、日本人が大切にしている根性とか精神力とかいう言葉。なぜか「倒れるまで頑張る」ことを美德と誤解している人が多い。玉砕しても負けは負けなのに、である。一方、西洋の精神力は「最後まで倒れない」ことをさす。なぜなら彼らは勝つために頑張るのだ。この違いは大きい。

ひいきの西ドイツは、今回のワールドカップで82年、86年に続いて決勝に進んだ。だが、これぞゲルマン魂という、逆境を勝ち抜く試合が今大会ではまだ見られない。それだけに決勝戦は波乱に満ちた展開を期待しよう。いまは決勝戦を2日前に控えた7月7日である。(T)

1990年9月号8月18日(土)発売

特集1 日本語を処理するために

特集2 2Dグラフィック続論

X68000にハンディスキャナをつなぐ

新製品紹介 ビデオボード/C compiler Ver.2.0(?)

新連載 清水和人流プログラミング道場/荻窪圭「大人のためのX68000」

全機種共通システム
ビリヤードゲーム

バックナンバー常備店

東京	神保町	三省堂神田本店5F 03(233)3312
	//	書泉ブックマートB1 03(294)0011
	//	書泉グランデ5F 03(295)0011
	秋葉原	T-ZONE 7Fブックゾーン 03(257)2660
	八重洲	八重洲ブックセンター3F 03(281)1811
	新宿	紀伊国屋書店本店 03(354)0131
	高田馬場	未来堂書店 03(200)9185
	渋谷	大盛堂書店 03(463)0511
	池袋	リプロ池袋店 03(981)0111
	//	西武百貨店9F コンピュータ・フォーラム 03(981)0111
神奈川	横浜	有隣堂横浜駅西口店 045(311)6265
	//	有隣堂ルミネ店 045(453)0811
	藤沢	有隣堂藤沢店 0466(26)1411

神奈川	厚木	有隣堂厚木店 0462(23)4111
	平塚	文教堂四の宮店 0463(54)2880
千葉	柏	新星堂カルチェ5 0471(64)8551
	船橋	リプロ船橋店 0474(25)0111
	//	芳林堂書店津田沼店 0474(78)3737
	千葉	多田屋千葉セントラルプラザ店 0472(24)1333
埼玉	川越	黒田書店 0492(25)3138
	川口	岩瀬書店 0482(52)2190
茨城	水戸	川又書店駅前店 0292(31)0102
大阪	北区	旭屋書店本店 06(313)1191
	都島区	駿々堂堂橋店 06(353)2413
京都	中京区	オーム社書店 075(221)0280
愛知	名古屋	三省堂名古屋店 052(562)0077
	//	パソコンΣ上前津店 052(251)8334
	刈谷	三洋堂書店刈谷店 0566(24)1134
長野	飯田	平安堂飯田店 0265(24)4545
北海道	室蘭	室蘭工業大学生協 0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は綴じ込みの振替用紙の「申込書」欄にある「新規」「継続」のいずれかに○をつけ、必要事項を明記のうえ、郵便局で購読料をお振り込みください。その際渡される半券は領収書になっていますので、大切に保管してください。なお、すでに定期購読をご利用の方には期限終了の

少し前にご通知いたします。継続希望の方は、上記と同じ要領でお申し込みください。

海外送付ご希望の方へ

本誌の海外発送代理店、日本IPS(株)にお申し込みください。なお、購読料金は郵送方法、地域によって異なりますので、下記宛必ずお問い合わせください。

日本IPS株式会社

〒101 東京都千代田区飯田橋3-11-6

☎03(238)0700



8月号

■1990年8月1日発行 定価560円(本体544円)

■発行人 孫正義

■編集人 橋本五郎

■発売元 ソフトバンク株式会社

■出版事業部 〒108 東京都港区高輪2-19-13 NS高輪ビル

Oh!X編集部 ☎03(5488)1309

出版営業部 ☎03(5488)1360 FAX 03(5488)1364

広告センター ☎03(297)0181

■印刷 凸版印刷株式会社

©1990 SOFTBANK CORP. 雑誌 02179-8 本誌からの無断転載を禁じます。

落丁・乱丁の場合はお取り替えいたします。

まるごと新作ボリュームアップ号

BEEP!

POWERFUL MEGA-MAGAZINE

MEGADRIVE

ビーブ!メガドライブ

8月号

定価480円(税込)
好評発売中

創刊1周年記念
ご愛読感謝!
222名プレゼント



〈特別付録〉
ヘルファイアー
特製ポスター

僕達のまわりに異星人がいっぱい
コンピュータゲームの宇宙人侵略史を探る

何かと話題のUFO特集

新作ラッシュの秋を目前に今後の展開を予想する

メガドライブの '90年後半戦を占う

とじこみ保存版

フェリオス攻略ガイド

ヘルファイアー スーパーモナコグランプリ
E SWAT バットマン 四天明王

ゲームボーイ専門誌 パワーアップした第2弾だ!

ゲームボーイLIFE VOL.2

定価380円(税込)

54本のソフトを総ガイド

輝け! 第1回ゲームボーイ大賞

試験にでないゲームボーイ講座

業界初の完全攻略 オールソーサリアンシリーズ

FALCOM MAGAZINE

[ファルコム・マガジン]

定価680円(税込)

オールアバウト・ソーサリアン パソコン版ソーサリアン
の総ガイドに加え、メガドライブ版ソーサリアンも紹介

オールファルコム・ベスト10 ファルコムユーザー100人
が選んだファルコムなんでもベスト10

ソフトバンク

ソフトバンクの 書籍特約書店

下記の書店の一覧は、ソフトバンク書籍特約店として右にある商品の他、新刊もとりそろえております。ご希望の商品がある場合は、下記のお近くの書店にてお買い求め下さい。

(注) 現品が売れて補充中の場合もございますので、ご注意下さい。



ソフトバンク出版事業部

〒108 東京都港区高輪2-19-13 ☎03(5488)1360

全国特約書店一覧

<p>〈北海道〉 札幌市 紀伊國屋書店札幌店 011-231-2131 旭屋書店札幌店 011-241-3007 丸善札幌支店 011-241-7252 リーブルなにか 011-221-3800 富貴堂札幌パルコ店 011-214-2303 ダイヤ書房本店 011-712-2541 ダイヤ書房西店 011-655-6223 旭川市 旭川富貴堂 0166-26-3481 ブックス平和マルカツ店 0166-23-6211 苫小牧市 旭屋書店苫小牧店 0144-36-5185 青森市 成田本店 0177-23-2431 岡田書店 0177-23-1381 弘前市 紀伊國屋書店弘前店 0172-36-4511 ブックイン城東 0172-28-2882 八戸市 伊吉書院 0178-44-1917 盛岡市 東山堂書店本店 0196-53-6464 さわや書店 0196-53-4411 第一書店 0196-53-3355 仙台市 金澤堂 022-225-6521 金澤堂ブックセンター 022-223-0979 アイエ書店駅前店 022-264-0718 丸善仙台支店 022-266-1127 高山書店 022-263-1511 ブックスみやぎ 022-267-4422 秋田市 三浦書店 0188-33-8131 山形市 八文字屋 0236-22-2150 福島市 岩瀬書店コルニエツタヤ店 0245-21-2101 博尚堂 0245-21-1161 郡山市 東北書店 0249-32-0379 いわき市 ヤマニ書房本店 0246-23-3481 鹿島ブックセンター 0246-28-2222 会津若松市 宝文館 0242-27-5198 原町市 文芸堂 0244-22-1720 〈関東〉 水戸市 川又書店駅前店 0292-31-0102 ツルヤブックセンター 0292-25-2711 勝田市 武石書店 0292-73-1212 東海村 大野書店 0292-82-2098 鹿島郡 なみき書店 0299-96-1855 土浦市 共栄堂 0298-21-6134 つくば市 丸善筑波大学会館店 0298-51-6000 友朋堂吾妻本店 0298-52-3665 宇都宮市 落合書店オリオン店 0286-34-3777 落合書店東武ブックセンター 0286-34-8271 新星堂宇都宮店 0286-33-2337 小山市 進駐駅ビル店 0285-25-1522 前橋市 煥平堂 0272-23-1211 リプロ前橋店 0272-34-1011 戸田書店前橋店 0272-61-5063 高崎市 学陽書房 0273-23-4055 サカキ書店 0273-62-1500 新星堂高崎店 0273-27-3961 戸田書店高崎店 0273-63-5110 太田市 ナカムラヤ 0276-22-2001 浦和市 須原屋本店 048-822-5321</p>	<p>浦和市 須原屋コソソ店 048-824-5321 大宮市 押田謙文堂 048-641-3141 ブックセンター押田 048-647-3141 三省堂ブックポート 048-646-2600 蕨市 須原屋蔵店 0484-44-1211 岩淵書店川口店 0482-52-2190 川口市 黒田書店川越店 0492-25-3138 所沢市 芳林堂所沢店 0429-25-5355 いけだ書店所沢店 0429-28-3271 上福岡市 黒田書店上福岡店 0492-66-0120 朝霞市 文教堂朝霞店 0484-76-0107 志木市 新星堂志木店 0484-74-0182 春日部市 文教堂春日部店 048-752-7666 比企郡 錦電サービス 0492-96-2962 千葉市 多田屋セントラルプラザ店 0472-24-1333 キティランド千葉店 0472-25-2011 習志野市 蔵聖堂 0474-72-5011 船橋市 ときわ書房本店 0474-24-0750 リプロ船橋店 0474-25-0111 旭屋書店船橋店 0474-24-7331 芳林堂津田沼店 0474-78-3737 第二蔵聖堂 0474-65-0926 三省堂書店西船橋店 0474-34-3111 西口アサノ 0471-44-2111 新星堂柏店 0471-64-8551 松戸市 堀江良文堂 0473-65-5121 辰正堂駅ビル店 0473-64-7997 横濱市 有隣堂トーヨー店 045-311-6265 有隣堂東ルミネ店 045-453-0811 栄松堂相鉄ジョイナス店 045-321-6831 そごうブックセンター 045-465-2111 丸善ブックメイソポルタ店 045-453-6811 有隣堂伊勢佐木店 045-261-1231 有隣堂戸塚店 045-881-2661 文華堂戸塚店 045-864-5151 アーバン文華堂 045-821-5151 文教堂青葉台南口店 045-983-5150 川崎市 有隣堂アゼリア店 044-245-1231 有隣堂川崎BE店 044-200-6831 文学堂本店 044-244-1251 文教堂溝ノロ店 044-811-8258 島森書店大船店 0467-46-3841 鎌倉市 鎌倉書店 0467-46-2619 平塚書房WALK店 0468-25-5537 有隣堂藤沢店 0466-26-1411 リプロ藤沢店 0466-27-0111 文教堂六会店 0466-82-9610 茅ヶ崎市 川上書店ルミネ店 0467-87-3827 平塚市 サクラ書店駅ビル店 0463-23-2751 文教堂四之宮店 0463-54-2880 小田原市 八小堂書店 0465-22-7111 伊勢治書店 0465-22-1366 文教堂小田原店 0465-36-3677 厚木市 有隣堂厚木店 0462-23-4111 大和市 文教堂中央林間店 0462-75-4165 相模原市 文教堂相模大野店 0427-49-0650 文教堂橋本店 0427-74-5581</p>	<p>相模原市 文教堂星ヶ丘店 0427-58-6121 津久井郡 文教堂城山店 0427-82-9278 千代田区 三省堂書店神田本店 03-233-3312 書泉グランデ 03-295-0011 東京堂書店 03-291-5181 旭屋書店水道橋店 03-294-3781 丸善お茶の水店 03-295-5581 蔵聖堂 03-291-1362 いずみ神田南口店 03-254-8521 明正堂秋葉原店 03-257-0758 T-ZONE 03-257-2660 中央区 八重洲ブックセンター 03-281-1811 日本橋丸善 03-272-7211 旭屋書店銀座店 03-573-4936 港區 書原新橋店 03-591-8738 雄峰堂N S店 03-503-6586 虎ノ門書房本店 03-502-3461 虎ノ門書房町店 03-454-2571 品川区 芳林堂大井町店 03-474-4346 明星書店五反田店 03-492-3881 渋谷区 紀伊國屋書店渋谷店 03-463-0131 旭屋書店渋谷店 03-476-3971 三省堂書店渋谷店 03-407-4545 大盛堂書店 03-463-0511 紀伊國屋書店笹塚店 03-485-0131 新宿区 紀伊國屋書店本店 03-354-0131 三省堂書店新宿西口店 03-343-4871 福家書店センタービル店 03-345-1246 福家書店野村ビル店 03-342-0298 新星堂N Sビル店 03-344-2055 西武新宿ブックセンター 03-208-0380 芳林堂高田馬場店 03-208-0241 未来堂 03-200-9185 豊島区 旭屋書店池袋店 03-986-0311 芳林堂池袋店 03-984-1101 リプロ池袋店 03-981-0111 三省堂書店池袋店 03-987-0511 新栄堂本店 03-984-2345 新栄堂アルパ店 03-988-0181 台東区 明正堂中通り店 03-831-0191 墨田区 ブックスア・談 03-635-1841 葛飾区 文教堂青戸店 03-838-5938 江戸川区 文教堂西葛西店 03-689-3621 大田区 アクトブックスサンカマタ店 03-735-1551 電文堂大森駅ビル店 03-775-3851 中野区 明星書店東京本社 03-387-8451 杉並区 ブックセンター荻窪 03-393-5571 書原杉並店 03-313-4778 武蔵野市 紀伊國屋書店吉祥寺東急店 0422-21-5543 弘栄堂吉祥寺店 0422-22-1031 バルコブックセンター吉祥寺 0422-21-8122 調布市 真光書店 0424-87-2222 府中市 啓文堂 0423-66-3151 三鷹市 三省堂書店三鷹店 0422-48-4510 東西書房 0422-46-0275 小金井市 文教堂小金井店 0423-86-0161 国分寺市 三成堂国分寺店 0423-25-3211</p>
--	--	---



展示図書一覧

定価は本体価格です。

MS-DOSいたれりつくせり本 ●1800円
 ブレイMS-DOS ●1900円
 UNIX System V
 プログラマ・ガイド ●12000円
 UNIX System V
 ユーザ・ガイド ●9800円
 UNIXオペレーティングガイド ●3000円
 OS/2 APIブックI ●2709円
 C言語の活用理解 ●2000円
 C言語の基礎知識 ●2500円
 C言語の応用50例 ●2300円
 上級・C言語の応用例50例 ●2400円
 Cプリプロセッサ・パワー ●2200円
 Play the C 上・下 ●各1500円
 Turbo C入門 ●2600円
 C++プログラミング ●2600円
 Quick Cプログラミング ●2602円
 詳説C言語 ●4369円
 8086アセンブリ言語 ●2800円
 8086マクロプログラミング ●2600円
 Final Ver.4.0ブック ●2400円

MIFES Ver.4.0ブック ●2400円
 ビジネスソフトデータ活用ブック ●2800円
 BASICによるプログラミング
 スタイルブック ●1800円
 ソーティング・ノート ●1900円
 J-3100パワーユーザーブック ●2400円
 続・PC工作入門 ●1800円
 PC-286Lブック ●1700円
 試験に出るX1 ●2800円
 RDBファラオ活用ガイド ●2903円
 言図ガイド ●2301円
 Rydeenガイド ●2427円
 P1EXEガイド ●2524円
 Lotus1-2-3ガイドII ●2500円
 MS-Chart Ver.3.1ガイド ●2900円
 まいと〜くガイド ●2300円
 新松ガイド ●2000円
 一太郎Ver.3ガイド ●2500円
 新一太郎ガイド ●2300円
 桐Ver.2ガイド ●2500円
 花子応用ガイド ●2500円

Lotus 1-2-3ガイド ●2400円
 P1ガイド ●2300円
 NinJa2 ガイド ●2300円
 Multiplan
 Ver.3.1ガイド ●2400円
 アセンブラCASL入門 ●2000円
 ハードウェア徹底マスター ●2500円
 FORTRAN徹底マスター ●2800円
 情報処理の基礎知識 ●1600円
 COBOL 徹底マスター ●2900円
 受験用語ハンドブック ●1800円
 情報処理入門1・2 ●各1204円
 CASLで学ぶ
 アセンブラ言語入門 ●2204円
 バイト&ワードの風について ●1800円
 田原総一郎のパソコンウォーズ ●1400円
 パソコンを襲う
 知的独占の戦い ●1600円
 RPG幻想事典・日本編 ●1800円
 魔法王国シムルグント ●1800円

国立市 東西書店 0425-75-5061
 小平市 文教堂小平店 0423-43-9229
 東村山市 文教堂東村山店 0423-96-1115
 立川市 オリオン書房ウエル店 0425-27-2311
 八王子市 くまざわ書店本店 0426-25-1201
 町田市 有隣堂町田店 0427-23-3018
 〃 久美堂本店 0427-25-1330
 〃 久美堂小田急店 0427-27-1111
 〃 文教堂鶴川店 0427-35-4117
 〃 文教堂小川店 0427-96-1781
 多摩市 くまざわ書店桜ヶ丘店 0423-37-2531
 福生市 文教堂福生店 0425-53-7708
 〈甲信越・北陸〉
 甲府市 文教堂甲府店 0552-22-4600
 長野市 平安堂長野店 0262-26-4545
 〃 長谷川書店 0262-26-2122
 上田市 平安堂上田店 0268-22-4545
 松本市 ブックスロクサン 0263-35-5555
 〃 改進黨松本駅前ビル店 0263-36-3777
 飯田市 平安堂飯田店 0265-24-4545
 岡谷市 笠原書店 0266-23-5070
 諏訪郡 平安堂下諏訪店 0266-28-1111
 新潟市 紀伊國屋書店新潟店 025-24-5281
 〃 萬松堂 025-229-2221
 〃 北光社 025-228-2321
 長岡市 寛張書店 0258-32-1139
 〃 ブックセンター長岡 0258-36-1360
 〃 長岡技大長峰文化 0258-46-6437
 上越市 バントピア コスモス 0255-25-5867
 山北町 BOOKメディア 0254-77-3850
 富山市 潮川書店 0764-24-4556
 〃 清明堂 0764-24-4166
 〃 BOOKSなかだ豊田店 0764-32-1353
 〃 文苑堂本郷店 0764-22-0552
 〃 文苑堂赤江店 0764-33-0321
 高岡市 文苑堂 0766-21-0333
 〃 文苑堂横田店 0766-21-0431
 金沢市 うつのみや片町店 0762-21-6136
 〃 書林香林坊本店 0762-20-5011
 野々市町 王様の本本店 0762-46-5325
 福井市 勝木書店 0776-24-0428
 〃 品川書店新田塚店 0776-24-1112
 〈東海〉
 静岡市 静岡谷島屋呉服町本店 0542-54-1301
 〃 江崎書店 0542-54-4481
 〃 吉見書店 0542-52-0157
 〃 戸田書店SBS店 0542-81-5733
 〃 戸田書店曲金店 0542-81-5899
 沼津市 吉野屋 0559-23-5676
 〃 マルサン書店宝塚店 0559-63-0350
 富士市 戸田書店富士店 0545-51-5121
 清水市 戸田書店本店 0543-65-2345
 浜松市 浜松谷島屋連尺本店 0534-53-9121
 名古屋市 三省堂書店名古屋店 052-562-0077
 〃 星野書店近鉄ビル店 052-581-4796
 〃 丸善名古屋支店 052-261-2251
 〃 丸善ブックメイソセントラルパーク 052-971-1231
 〃 日進堂上前津店 052-263-0550

名古屋市中 三洋堂パソコンショップ2 052-251-8334
 〃 三洋堂いりなか本店 052-832-8202
 〃 ちくさ正文館本店 052-741-1137
 〃 白樺書房西店 052-774-7223
 豊橋市 精文館 0532-54-2345
 岡崎市 ブックス鎌倉 0564-54-1822
 豊田店 三洋堂梅坪店 0565-35-2334
 〃 三洋堂豊川店 05338-3-0334
 刈谷市 三洋堂刈谷店 0566-24-1134
 春日井市 三洋堂豊川店 0568-32-7806
 岐阜市 自由書房 0582-65-4301
 大垣市 大洞堂ブックス258 0584-81-2553
 〃 大洞堂岐阜大バイパス店 0584-74-7766
 一宮市 三洋堂一宮店 0586-77-5734
 多治見市 三洋堂多治見店 0574-63-2334
 〃 三洋堂多治見店 0572-24-0340
 津市 別所書店11ビル店 0592-24-1014
 四日市市 文化センター白揚 0593-51-0711
 鈴鹿市 シェフ白揚スズカ 0593-82-5221
 〈近畿〉
 京都市 駿々堂京宝店 075-223-1003
 〃 アバンティ・ブックセンター 075-682-5031
 〃 オーム社書店河原町店 075-221-0280
 〃 ジュンク堂京都店 075-252-0101
 〃 オーム社書店竹田店 075-644-2611
 奈良市 駿々堂大丸店 0742-26-6241
 大阪市 旭屋書店本店 06-313-1191
 〃 紀伊國屋書店梅田店 06-372-5821
 〃 オーム社書店大阪店 06-345-0641
 〃 駿々堂京橋店 06-353-3209
 〃 駿々堂心斎橋店 06-251-0881
 〃 旭屋書店ナンパ店 06-644-2551
 〃 ナンパブックセンター 06-644-5501
 〃 ヒバリヤ書店ナンパ店 06-644-5407
 〃 旭屋書店アペノ店 06-631-6051
 〃 ユーゴー書店 06-623-2341
 〃 河村書店 06-951-2968
 枚方市 水嶋書房京阪デパート店 0720-51-3432
 高槻市 コーベックス西武高槻店 0726-83-1766
 東大阪市 ヒバリヤ書店本社 06-722-1121
 神戸市 ジュンク堂センター街店 078-392-1001
 〃 ジュンク堂サンパル店 078-252-0777
 〃 海文堂書店 078-331-6501
 〃 日東館書林 078-391-8701
 姫路市 新舞書房 0792-85-3344
 〃 誠心堂書店 0792-81-2055
 和歌山市 宮井平安堂 0734-31-1331
 〃 帯伊書店 0734-22-0441
 〈中国〉
 岡山市 紀伊國屋書店岡山店 0862-32-3411
 〃 金明堂書店 0862-31-2261
 津山市 津山ブックセンター 08682-6-4047
 広島市 紀伊國屋書店広島店 082-225-0332
 〃 丸善広島支店 082-247-2251
 〃 金正堂 082-248-3715
 〃 積善館 082-248-3151
 尾道市 啓文社尾道店 0848-37-5151
 福山市 啓文社福山店 0849-22-3111

福山市 ブックシティ啓文社 0849-25-0050
 〃 啓文社コア 0849-41-0909
 山口市 五十部誠文堂 0839-24-6630
 〃 文栄堂 0839-22-5611
 下関市 中野書店 0832-22-5181
 宇部市 京屋書店 0836-31-2323
 〃 不広書店 0836-31-0086
 防府市 誠文堂国街店 0835-25-1988
 〃 光市 三文字屋 0833-71-0251
 鳥取市 富士書店 0857-23-7271
 松江市 園山書店 0852-21-4167
 〈四国〉
 徳島市 小山助学館本店 0886-54-2135
 〃 小山助学館東口店 0886-25-1380
 〃 森住九善 0886-23-3228
 高松市 宮船書店本店 0878-51-3733
 丸亀市 宮船書店丸亀店 0877-22-5533
 松山市 紀伊國屋書店松山店 0899-32-0009
 〃 明星書店本店 0899-41-4141
 〃 明星書店大街道店 0899-41-4242
 〃 九三書店 0899-31-8501
 新居浜市 明星屋原店 0897-44-4000
 宇和島市 明星宇和島店 0895-23-1118
 高知市 金高堂 0888-22-0161
 〈九州・沖縄〉
 福岡市 紀伊國屋書店福岡店 092-721-7755
 〃 リーブル天神 092-713-1001
 〃 横文館新天町店 092-781-2991
 〃 福岡金文堂本店 092-741-2106
 〃 福岡金文堂朝日ビル店 092-431-1094
 〃 福岡金文堂デイトス店 092-451-6175
 〃 福岡金文堂アニメート原 092-844-0088
 北九州市 ナガリ書店 093-521-1044
 〃 金栄堂 093-531-3685
 〃 旭屋書店北九州店 093-631-6421
 〃 井筒屋ブックセンター 093-641-0131
 〃 カルパーク平野 093-661-7988
 〃 白石書店本城店 093-601-2200
 久留米市 エマックスたがみ 0942-33-1841
 飯塚市 BOOKリード 0948-25-7266
 大分市 バルコックセンター大分店 0975-35-0643
 〃 本町晃星堂 0975-33-0231
 別府市 明林堂 0977-23-2183
 宮崎県 中央・田中書店 0985-24-3511
 〃 寿屋宮崎店 0985-27-4111
 佐賀市 金屋堂北バイパス店 0952-32-1965
 〃 横文館佐賀店 0952-24-4314
 〃 横文館デイトス店 0952-23-7155
 長崎市 メトロ書店 0958-21-5453
 〃 好文堂 0958-23-7171
 佐世保市 金明堂書店 0958-22-4214
 熊本県 紀伊國屋書店熊本店 096-322-5531
 〃 長崎書店 096-353-0555
 人吉市 明星人吉店 0966-22-5486
 鹿児島市 春苑堂ブックプラザ 0992-25-3200
 〃 ブックスみずみ 0992-57-1011
 那覇市 球陽堂書房ビル店 0988-63-3752
 〃 文教堂 0988-62-1201

大学生のための パソコン業界研究セミナー

(平成2年) 7月1日

SOFTBANK NEWS

ソフトバンク株式会社
〒108 東京都港区高輪2-19-13 NS高輪ビル
TEL: 03-5488-1115

LAN業界大手企業との 合併事業大規模展開

ソフトバンクは米国最大のLANソフト専門会社、ノベル社と提携。ノベル株式会社会を足元させた。米国より運来している日本

LAN市場が大きく変動する可能性がある。米国より運来している日本

LANの普及が速く、ノベル社の代理店が、ノベル社と提携

LANの普及が速く、ノベル社の代理店が、ノベル社と提携

LANの普及が速く、ノベル社の代理店が、ノベル社と提携

LANの普及が速く、ノベル社の代理店が、ノベル社と提携

九十年代新卒入社社員 集合研修開催

フレッシュな新人が夢と希望を抱いて、四月二日より八日間に

フレッシュな新人が夢と希望を抱いて、四月二日より八日間に

フレッシュな新人が夢と希望を抱いて、四月二日より八日間に

フレッシュな新人が夢と希望を抱いて、四月二日より八日間に

フレッシュな新人が夢と希望を抱いて、四月二日より八日間に

フレッシュな新人が夢と希望を抱いて、四月二日より八日間に

フレッシュな新人が夢と希望を抱いて、四月二日より八日間に

PC WEEK

日本語版創刊

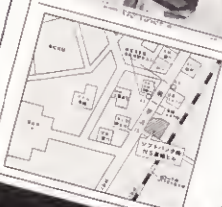
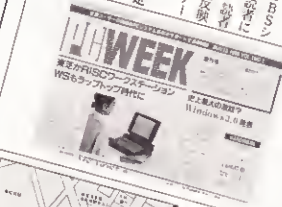
ソフトバンク出版事業部、四月十八日、パソコン

ソフトバンク出版事業部、四月十八日、パソコン

ソフトバンク出版事業部、四月十八日、パソコン

ソフトバンク出版事業部、四月十八日、パソコン

ソフトバンク出版事業部、四月十八日、パソコン



セミナー開催日決定!

ソフトバンク株式会社セミナー
熱烈待望! 電界講習!

ソフトバンク株式会社 パソコン業界研究セミナー日程

会場	連絡先	会場	連絡先
札幌: 7月23日	TEL 011 (222) 6026	札幌営業所	
仙台: 7月20日	TEL 022 (263) 0907	仙台営業所	
東京: 7月16、24日	TEL 03 (5488) 1115	本社人事部	
名古屋: 7月26日	TEL 052 (261) 7215	名古屋営業所	
大阪: 7月 9日	TEL 06 (264) 1471	西日本営業部	
広島: 7月11日	TEL 082 (223) 1314	広島営業所	



ソフトバンク
〒108 東京都港区高輪2-19-13
NS高輪ビル
TEL 03 (5488) 1115

HOST

△68000 専用
多回線 ホストソフト

PRO-68K

ついに
登場!

3回線 / 9回線

きみも、今日から局長さん

HOST 9 PRO-68K 概要

対応回線数 1~9回線
使用モデム ATモデム MNP(RTS/CTS)可
通信速度 最大9600bps
会員数 *最大9999人
掲示板数 *最大40個
機能 電子掲示板・電子手帳・電子会議(チャット)・会員情報

これらは、コンフィグファイルで設定できます。
注1: *印について拡張を希望する場合は、プログラムの書き換えが必要になりますので、別料金にて対応致します。当社までご相談ください。
2: 2回線以上で運用される場合は、CZ-68F1(シャープ純正)が必要になります。
3: このホストはテキスト形式の転送方法を採用しております。

■特長

- 各種設定のコンフィグファイル化。●RS-232C回線とは別にキーボードからのアクセス、ダウンロード、アップロードが可能。●モニターで、各チャンネルのユーザーの打ち込んだコマンドや通信状態を確認。●各掲示板別にSIG、ボードパスの設定。●メンテナンス作業のオンライン実行。(ボードメンテナンス、メールメンテナンス)●オンラインサインアップ等、ゲストへの設定が可能。
- 通信サービスTNP-P対応。●行編集(オンライン簡易エディタ)機能。●その他、システムレベルで会員情報の変更が可能。タイムアウトによる回線切断。PDS専用掲示板の採用。(1書込中で、ドキュメントとテキストプログラムの分離)。●接続MNPタイプの識別。●ログイン、ログアウト時間の記録。●非アクセス時のモニター画面消去可能。

HOST 3 PRO-68K

機能は統べて、「HOST 9 PRO-68K」と同じですが、対応回線数が、1~3回線に制限されて、低価格でユーザーに供給します。

バージョンアップ (Ver1.10) サービス実施中

現在発売されています製品は、Ver1.10に変更になっています。お使いの製品がVer1.00のユーザーの方のために、バージョンアップサービスを実施しておりますので、お早目に、ユーザー登録葉書をお送り下さい。
Ver1.10へ無料交換を実施しております。

好評発売中

HOST 9 PRO-68K ¥59,800円

HOST 3 PRO-68K ¥39,800円

SPS-NET
TSUKUMO-NET モデル運用中!!

今、X68000の
通信が変わる!!!

ユーザー情報の機能を拡張して
好評発売中
17,800円
24/31KHz
ディスプレイ
対応

た〜みのる

2

「た〜みのる」が
装いも新たに
「た〜みのる2」として登場!
「た〜みのる」が
通信人門版なら
「た〜みのる2」は
マニアタイプの通信ソフトです!!!

△68000 専用
パソコン通信ソフト

プログラマ募集!!

SPSでゲームを作ってみませんか?

アセンブラでプログラムの組める優秀な人材を若干名募集しています。就職希望の方は62円切手同封の上、「就職案内係 大和」までお手紙ください。折り返し就職のご案内をお送り致します。

尚、デザイナー、音楽プログラム等の専門職は募集しておりません。



(株)マイコンハウス
SPS

当社の製品は全国の有名デパート、パソコンショップやお求めになります。尚、お求めにならない場合、郵便物にてお申し込みください。●口座番号 8515-12298
●加入者名 株式会社マイコンハウス ●金額 代金に3%の消費税を加算した額 ●通信料 (裏面)ご希望ゲームソフト名、数量、代金合計、年齢、氏名、機種名、テープかディスクの種類、(一週間以上かかりますので、お急ぎの方は現金書留をご利用ください。その場合、おつりのいらぬようにお願いします。

■表示価格に消費税は含まれておりません。

△68000
HOST PRO-68K 使用

SPS-NET TEL (0245)46-1167代

好評/一般回線
Tri-P (4回線) / (9回線) MNPクラス7

24時間運営 (N81XN)
ゲストID (GUEST)

*GUESTアクセスは無料です。ぜひ、一度試してください。

例◎パスワード=SPS-NET (8文字まで大小文字の混在可) ◎職業=株式会社エス・ビー・エス(16文字まで)
◎本名=大和太五郎(8文字まで) ◎住所=福岡市太平寺町ノ内5-3(24文字まで)
◎ペンネーム=大ちゃん(4文字まで) ◎自己紹介=SPS-NETをよろしく(24文字まで)
◎年齢=30(現在の年齢) ◎システム構成=X68000ACE-HD MD2400B(18文字まで)
◎電話=0245-45-5777(市外局番から)

入会方法 登録料¥3,000(税別) 会費無料

下記の用紙に直接記入するか又は、コピーして記入し、72円切手同封の上、「SPS-NET係」までお送り下さい。届き次第、仮登録を行いID発行後SPS-NET専用の郵便振込み用紙ならびに運用の手引きをお送りいたします。それに従い、3ヶ月以内に登録料3,000円(税別)を御入金下さい。入金確認後正式会員として再登録します。



専用

turbo OK-システム 漢字

「個人簿記会計 財計くん」2HD版
定価 49,800円 (税別)

出力帳票：勘定科目一覧表・摘要一覧表・期首貸借対照表・期末試算表・貸借対照表・損益計算書・仕訳帳・各科目別元帳・合計残高試算表

処理金額 9桁 10億円/年間
月間仕訳処理数 900件以内
仕訳入力は一度 振替伝票方式採用
使用勘定科目数 75個(年度変更可)
摘要小書き入力 A・Bの2つ
Aはコード入力
Bは自由入力
オート・ソート 仕訳訂正で
日付自動処理
ラクラク金額入力 カンマ付き、無
どちらもOK!

消費税の会計処理 注目の消費税の
会計処理は、4つの対応が考えられ
ますが、ユーザー別に勘定科目の設
定をする事により処理できます。

「消費税検証」を別冊にて同梱し
てあります。ご活用下さい。

〈各種税法は変化しても、複式簿記
の原理は不変です。勘定科目の設定
によって処理できるのが、財計くん
なのです。〉

プリンター用紙

縦11イン치의白紙又は野線入りを
使用願います。

2D版との能力アップの内容

1. ディスクの入れ替えなしで、システム・ユーザー辞書使用可。
2. 科目&摘要の入力時にHTLPキー機能を追加。

「個人簿記会計 財計くん」2D版
定価 39,800円 (税別)

2HD版との相違は、先の能力アップの内容の通りです。

各資料のご請求は

資料は、一部あたり200円分の切手
を同封願います。各デモ・サンプル
版は実費2400円を申し受けます。

弊社へ直接お申込みの方は上記分
を差し引いてご本体を購入できます。

資料は毎月曜日に、デモ版は逐次
発送しています。

「財計くん 売掛管理台帳」2HD版
定価 39,000円 (税別)

出力帳票：納品書・請求書・アイウエ
オ順顧客一覧表・取扱商品一覧表・
売上日計表・売掛残高一覧表・DM
シール(条件検索可)

処理金額 9桁 10億円/年間
1顧客処理件数 60件/月間 繰越可
処理顧客数 1DataDisk 1200名
取扱商品数 1DataDisk 250品目
消費税自在処理 登録済使用と未登録
使用どちらも可
登録済顧客変更 台帳変更Bで自在
帳票3段階選択 顧客別&メ切&全部
商品単価無登録 250品目が無限に
ラクラク金額入力 カンマ付き、無
どちらもOK!

プリンター対応表

ご使用になる機種により4つのシ
リーズ品番がございます。ご購入の
際にはご確認願います。

No701:CZ-8PK3・CZ-8PK4・CZ-
8PK5・C-8PK6・CZ-8PK7・

CZ-8PK8・CZ-PK9・EPSO
N-VPシリーズ=X1ROM要

No702:CZ-8PK2・CZ-80PK

No703:CZ-8PD2・CZ-8PD2・CZ-
800P・EPSON-SPシリーズ
=X1ROM要

No704:X1に接続可能なもので、縦
11イン치의白紙又は野線入り
のもののみを利用する事になり
ます。

* 伝票専用用紙として、ヒサゴ(株)
GB-342を使用します。伝票以外
は縦11イン치의連続用紙(白紙or
野線入り)を使います。なお、No.
704のみは、伝票用紙はユーザーが
作成して使用する事になります。

2D版との能力アップの内容

1. ディスクの入れ替えなしで、システム・ユーザー辞書使用可。
2. 商品名の入力時にHELPキー機能が追加。

「財計くん 売掛管理台帳」2D版
定価 29,000円 (税別)

2HD版との相違は、先の能力アップ
の内容の通りと、処理顧客数が600
名となり、取扱商品数が150品目と
なります。(2HD同様No701~No.
704品番がございます。ご購入の際
はご確認下さい。)

「DATA・CARD 1200」2HD版
定価 42,000円 (税別)

カード型データベースとしての
機能とグラフ作成ツールのグラフデー
ター・ファイル機能を持っています。
検索は、1,124枚のデータカード
から3重条件を処理します。

項目設定は自由設定で12個までを
処理し、データ部は新規に設けま
した「データ変換Uty」で、作
成済みのデータでもデータ量に
応じて変更可能になりました。

DMシール発行・葉書宛名印刷を
条件検索で処理します。

カードNoによる、データの抜
粋・ステップ印刷(同カードを最大
12枚まで)を処理します。

グラフ・ツールとしては、7種・
22タイプのグラフを作成する事がで
き、最大12項目12データを縦棒グ
ラフ・横棒グラフ・帯グラフ・円グ
ラフ・折線グラフに処理します。縦
棒グラフ・横棒グラフは3D仕様で
も処理します。

プリンター用紙

縦11イン치의白紙又は野線入りを
使用願います。

2D版との能力アップの内容

1. ディスクの入れ替えなしで、システム・ユーザー辞書使用可。
2. グラフDataDisk内に格納できるファイル数が3倍になりました。

「DATA・CARD 1200」2D版
定価 32,000円 (税別)

2HD版との相違は、先の能力アップ
の内容の通りです。

ご購入は

お近くのパソコン・ショップでお
求め下さい。お急ぎの方は直接現金
書留でお申し込み下さい。

(売掛管理台帳のNo704のみユーザー
のご希望により、プログラム解放型
2D¥58,000円(税別)もあります。
直接弊社にお申し込みください。)

〒885 宮崎県都城市都島町430-2

OK-ハウス

TEL 0986-25-0303-FAX 0986-25-9555

秋葉原価格でローンができます
電気街秋葉原で
25年
の信用!!

AVCフタバ

03(253)7661
〒101 東京都千代田区外神田3-2-3 ☎03-253-7611(代)



今すぐ もよりの電話から	仙 台 022-264-3704	名古屋 052-452-3271	広 島 082-295-6873
札 幌 011-611-5104	新 潟 0252-75-4175	大 阪 06-311-3931	福 岡 092-481-2494

X68000の情報のすべて! (当店はX68000の認定代理店です。お気軽にご相談下さい)

68000 待望の新しい仲間登場!!

PERSONAL WORKSTATION
EXPERT II・EXPERT II HD

EXPERT II・EXPERT II HD
集積度を高めた"マンハッタンシェイプ"3Mの大容量メモリを搭載。本格的なウィンドウシステム、SX-WINDOW搭載。

(写真のモニターは別売です。)

CZ-603C 標準価格 ¥338,000
CZ-613C 標準価格 ¥448,000

AVC 特価

68000

PERSONAL WORKSTATION
PRO II・PRO II HD

PRO II・PRO II HD
拡張I/Oポートを4スロットを搭載し、汎用性と低価格が魅力。もちろん、SX-WINDOW搭載。

CZ-653C 標準価格 ¥285,000
CZ-663C 標準価格 ¥395,000

AVC 特価

X68000

在庫稀少価格はお電話で!

CZ-602C
CZ-612C
CZ-652C
CZ-662C

お勧めディスプレイコーナー 組合せは自由、価格はお気軽にご相談下さい。

<p>CZ-604D 標準価格 ¥94,800 AVC 特価</p> <ul style="list-style-type: none"> ●0.31mmドットピッチ ●2モードオートスキャン ●ステレオスピーカー搭載 ●チルト台同梱 	<p>CZ-613D 標準価格 ¥135,000 AVC 特価</p> <ul style="list-style-type: none"> ●ドットピッチ 0.31mm ●TVチューナー搭載 ●ステレオスピーカー搭載 ●チルト台同梱 	<p>CZ-603D 標準価格 ¥84,800 AVC 特価</p> <ul style="list-style-type: none"> ●0.31mmドットピッチ ●TVチューナー無し ●3モードオートスキャン ●チルト台同梱
<p>CU-21HD 標準価格 ¥148,000 AVC 特価</p> <ul style="list-style-type: none"> ●0.52mmドットピッチ ●21型ディスプレイ ●3モードオートスキャン ●ステレオスピーカー搭載 	<p>CZ-605D 標準価格 ¥115,000 AVC 特価</p> <ul style="list-style-type: none"> ●ドットピッチ 0.39mm ●TVチューナー搭載 ●ステレオスピーカー搭載 ●チルト台同梱 	<p>CZ-602D 標準価格 ¥99,800 AVC 特価</p> <ul style="list-style-type: none"> ●ドットピッチ 0.39mm ●TVチューナー搭載 ●チルト台同梱

型 番	品 名	標準価格	販売価格	型 番	品 名	標準価格	販売価格	型 番	品 名	標準価格	販売価格
CZ-6TU	システムチューナー	¥ 33,100	AVCフタバ特価	CZ-8PG1	24ピンカラープリンター (80列)	¥ 130,000	AVCフタバ特価	CZ-8TM2	モデムユニット	¥ 49,800	AVCフタバ特価
BF-68PRO	CRTフィルター	¥ 19,800	AVCフタバ特価	CZ-8PK1D	24ピンプリンター (136列)	¥ 97,800	AVCフタバ特価	CZ-252MS	Musicstudio	¥ 28,800	AVCフタバ特価
CZ-8NS1	カラーキャプチャー	¥ 188,000	AVCフタバ特価	IO-735X	カラージェットプリンター	¥ 248,000	AVCフタバ特価	CZ-247MS	MUSIC (MID)	¥ 28,800	AVCフタバ特価
CZ-6BN1	スキャナー用パラレルボード	¥ 29,800	AVCフタバ特価	CZ-6BE1A	1M増設RAMボード	¥ 38,000	AVCフタバ特価	CZ-221HS	NEW Print Shop	¥ 19,800	AVCフタバ特価
CZ-6VT1	カラーイメージユニット	¥ 69,800	AVCフタバ特価	CZ-6BE2	2M増設RAMボード	¥ 79,800	AVCフタバ特価	CZ-228BS	TOP給与計算エクスパート	¥ 200,000	AVCフタバ特価
CZ-8BV2	カラーイメージボード	¥ 39,800	AVCフタバ特価	CZ-6BP1	4M増設RAMボード	¥ 138,000	AVCフタバ特価	CZ-227BS	TOP財務会計	¥ 200,000	AVCフタバ特価
CZ-8BR1	立体映像セット	¥ 29,800	AVCフタバ特価	CZ-6BM1	数値演算プロセッサ	¥ 79,800	AVCフタバ特価	CZ-220BS	DATA	¥ 58,000	AVCフタバ特価
CZ-8DT2	パーソナルテロップ	¥ 44,800	AVCフタバ特価	CZ-6BC1	FAXボード	¥ 79,800	AVCフタバ特価	CZ-212BS	BUSINESS	¥ 68,000	AVCフタバ特価
CZ-8BS1	FM音源ボード	¥ 23,800	AVCフタバ特価	CZ-6BL1	MIDIボード	¥ 26,800	AVCフタバ特価	CZ-219SS	OS-9	¥ 29,800	AVCフタバ特価
CZ-8NJ1	ジョystick	¥ 1,700	AVCフタバ特価	CZ-6BL1	I/Oボード	¥ 39,800	AVCフタバ特価	CZ-211LS	Ccompiler	¥ 39,800	AVCフタバ特価
CZ-8NM2A	マウス	¥ 6,800	AVCフタバ特価	CZ-243BS	LANボード	¥ 268,000	AVCフタバ特価	CZ-234LS	AI-68K	¥ 188,000	AVCフタバ特価
CZ-8NM3	マウス・トラックボール	¥ 9,800	AVCフタバ特価	CZ-240BS	サイバーノート	¥ 19,800	AVCフタバ特価	CZ-620H	20MBハードディスク	¥ 178,000	AVCフタバ特価
CZ-6SD1	システムラック	¥ 44,800	AVCフタバ特価	CZ-223CS	ステーションリレー	¥ 14,800	AVCフタバ特価	CZ-64H	40MBハードディスク	¥ 120,000	AVCフタバ特価
AN-S100	アンブレ内蔵スピーカー	¥ 36,600	AVCフタバ特価	CZ-223CS	通信ソフト	¥ 15,800	AVCフタバ特価	LHD-34V	40MBハードディスク (縦向き)	¥ 153,000	¥ 117,000
CZ-6EB1	拡張I/Oボックス	¥ 88,000	AVCフタバ特価		ゲームソフト	20% OFF		LHD-32V	20MBハードディスク (縦向き)	¥ 128,000	¥ 98,000

CZ-8NJ2

アナログジョystick
標準価格 ¥23,800

AVC 特価 ¥???

CZ-8PG2

24ピンカラー。
漢字ドットインパクト
プリンター

CZ-8PG2.....¥160,000

AVC 特価 ¥???

CZ-8PC4

48ドット熱転写プリンター。
精密な文字、ハード
コピーも可能。

CZ-8PC4.....¥ 99,800

AVC 特価 ¥64,800

CZ-8PC3

24ドット熱転写カラー
プリンター

標準価格.....¥ 55,800

AVC 特価 ¥39,800

●頭金なし(手軽な電話クレジット) ●製品先取り(お支払いは約1~2ヶ月後から) ●低金利クレジット(1回の支払いは2,700円以上で3~48回。ボーナス併用可) ●クレジットクレジット(保証人なし。但し満20歳以上の学生の方) ●18歳未満の方(ご両親が代理購入者としてお申し込み下さい) ●納期(通常の場合、当社に申込書が到着後1週間以内。特に人気のある商品で品薄の場合、少々納期が遅れることがありますのでご了承下さい) ●完全保証(すべてメーカー保証書付。アフターケア万全) ●全国代引(お届けした者に、代金をお支払いいただく方法です。但し手数料1,000円)

AM10時からPM7時
まで受付 日曜・祝日も営業

信用と実績を誇る

BASIC HOUSE

BASIC HOUSE 大田原店

一周年

大謝恩セール!

〈主な取扱いメーカー〉

Apple Computer

FUJITSU

NEC

EPSON

SHARP

TOSHIBA

全品大特価!!

7/20(金)~7/23(月)

BASIC HOUSE 宇都宮

店内改装のため

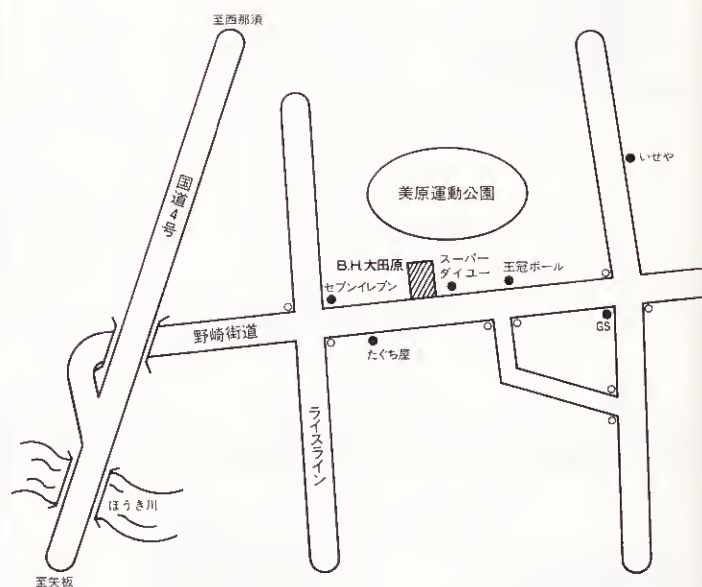
★閉店売りつくしセール

7/20(金)~7/23(月)

★新装開店セール

8/3(金)~8/6(月)

※通信販売もOKです!!



全国どこでも発送可 長期クレジットOK 送料全国均一¥1,000 宅配便にて即日配送

株式会社計測技研

本社営業部/マイコンショップ/通販部
大田原営業所/マイコンショップ

宇都宮市竹林町503-1 TEL.0286-22-9811 FAX.0286-25-3970
大田原市美原1-13-4 TEL.0287-23-5352 FAX.0286-23-5364

マイコンショップ **BASIC HOUSE**

お申し込み・お問い合わせは **☎0286-22-9811(代)**

2枚のボードが1枚になった

KGB-X68PRK



絶賛発売中

広大なメモリ空間を実現する最大4Mバイトの
高速増設メモリ

高速演算を約束してくれる
**数値演算
プロセッサ**

- メモリアクセスノーズウェイトによる高速アクセス
- CZ-6BE2/CZ-6BE4/CZ-6BP1との混在が可能です
- 複数枚のKGB-X68PRKの実装が可能です
- ジャンパの変更により任意のアドレス空間にメモリの配置が可能です
- ジャンパの変更により数値演算プロセッサの1枚目2枚目 未使用の選択が可能です
- 1M/2M/3Mメモリモデルは購入後にメモリをボード上に追加可能です
- 数値演算プロセッサにはデバイスドライバ(FLOAT3X)が付属します

※写真はKGB-X68PRK-14です

※CZ-602C/CZ-612C以外の機種ではCZ-6BE1/CZ-6BE1Aを実装している必要があります
※メモリアクセスノーズウェイトのため拡張 I/O BOXでは動作しません

製品価格一覧

KGB-X68PRK-01	¥ 58,000
(1Mメモリ/数値演算プロセッサ無し)	
KGB-X68PRK-02	¥ 74,000
(2Mメモリ/数値演算プロセッサ無し)	
KGB-X68PRK-03	¥ 98,000
(3Mメモリ/数値演算プロセッサ無し)	
KGB-X68PRK-04	¥122,000
(4Mメモリ/数値演算プロセッサ無し)	

KGB-X68PRK-11	¥ 96,000
(1Mメモリ/数値演算プロセッサ付き)	
KGB-X68PRK-12	¥112,000
(2Mメモリ/数値演算プロセッサ付き)	
KGB-X68PRK-13	¥136,000
(3Mメモリ/数値演算プロセッサ付き)	
KGB-X68PRK-14	¥160,000
(4Mメモリ/数値演算プロセッサ付き)	

購入後の増設費用

メモリ	
1Mバイト	¥24,000
2Mバイト	¥51,000
3Mバイト	¥76,000
数値演算プロセッサ	
MC68881RC16	¥38,000

充実のBASIC HOUSEハードウェア&ソフトウェア

高速12BIT, 16CH A/Dコンバータボード(KGB-AD12) X1	¥118,000
フォトアイソレーション16BITデジタル入出力ボード(KGB-PIO) X1	¥ 42,000
ハードディスクインターフェースボード(KGB-HDIF) X1	¥ 16,000
アイソレーション16BITデジタル入出力ボード(KGB-X68PIO) X68000	¥ 68,000
ハンディプリンタ&インターフェース(HANDYPRINTjack) X68000	¥ 24,800

高速12BIT, 4CH D/Aコンバータボード(KGB-DA4) X1	¥ 98,000
汎用ローコストA/D&PIOボード(KGB-XIS) X1	¥ 19,800
高速12BIT, 16CH A/Dコンバータ(KGB-X68ADC) X68000	¥128,000
64180CPUボードMach180(KGB-CPXB) X68000	¥ 98,000
ローコストMIDIインターフェース(MELODY BOX) X68000	¥ 15,800

BASIC拡張関数パッケージ(B6-6301)	¥9,800
C言語ライブラリ(B6-6305)	¥6,800
ディスクキャッシュ(B6-6304)	¥6,800
Toys & Tools (B6-6307)	¥6,800

BASIC拡張関数パッケージC言語ライブラリ付(B6-6306)	¥ 14,800
アイコンエディタ(B6-6303)	¥4,800
CP/M5.0エミュレータ(B6-6302)	¥ 19,800

PRKニューバリエーション販売開始! PRK10コプロセッサ付/メモリー無し 定価¥72,000

BASICHOUSE BBS TECOSYS NET

TEL 0286-27-1829 / 1200/2400ボー・MNPクラス5 8ビット/パリティ無し/X制御無し
ゲストIDなし(オンラインサインアップ)

全国どこでも発送可 長期クレジットOK 送料全国均一¥1,000 宅配便にて即日配達

株式会社計測技研

本社営業部/マイコンショップ/通販部
大田原営業所/マイコンショップ

宇都宮市竹林町503-1 TEL 0286 22 9811 FAX 0286 25 3970
大田原市美原1-13-4 TEL 0287-23 5352 FAX 0286 23 5364

マイコンショップ BASIC HOUSE

お申し込み・お問い合わせは ☎0286-22-9811(代)

アイ・ツ- EXE CLUB

新規ユーザー・EXE会員 大集合

- ★ X68000ユーザーニーズに対応したハード・ソフト・ウエア・周辺機器は全て展示しています。
- ★ 新製品情報・ユーザー同士の情報交換ができる、メンバー様の憩いのスペースです。
- ★ 大特価セール期間中X68000・ディスプレイ・プリンター御購入の方は全国どこでも送料無料!!
- ★ 遠くでなかなかお越し頂けない方にも通販専用TELで専門スタッフ(X68 PRO STAFF)が親切丁寧に答えします。
- ★ X68000お買い上げの方、アイ・ツ-よりBigプレゼント。

X68000 オリジナルステッカー
X68000 フロッピータイトルシール
X68000 オリジナルテレホンカード
X68000 バッグ

- ★ 現在シャープX68000 EXE会員の方、おトモたちをご紹介下さい。ご購入成立時点でアイ・ツ-とシャープよりステキなプレゼント進呈中!!

- ★ アイ・ツ-メンバーズ優待制度実施
アイ・ツ-でX68000及びソフトウエア周辺機器をお買い上げ頂きましたユーザー様にはオリジナルメンバーズカードを送付致します。メンバーの方には楽しいパソコンライフをおくれますように最善のフォローをアイ・ツ-より提供します。

新製品入替機

展示品処分祭 早い者勝ち

EXPERT SET CZ-612CBK
CZ-603DBK

定価+3%=¥567,324

大放出!!

40% off

あなたの
お支払は
タツタの

¥338,000
ポツ・キリ

シャープ販売コンテスト・パソコン部門最高峰賞 シャープ販売第一位受賞感謝セール! ■期間/7月18日~8月17日

X68000プロショップ(専門店) ならではの企画です!!

SX-WINDOW ってなあ~に?

X68000ユーザーみんな集まれ!
SX-WINDOWの勉強会?を
開催します。

参加ご希望の方は、62円切手同封のうえ、お名前・ご住所・TEL・生年月日・お持ちのX68000の型番を書いて、
アイ・ツ-EXE CLUBあてで、おくって下さい。
日時、場所etc...ご連絡します!!

場所はとりあえず大阪です!

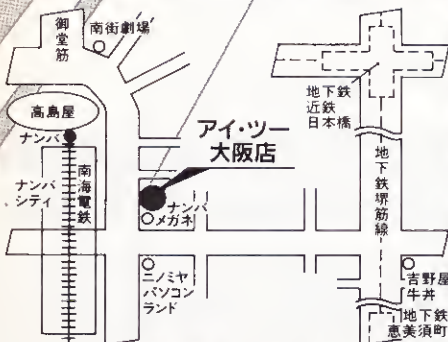
チャンスです
逃がす手はない

通販専用TEL.

06-633-9800

アイ・ツ-inシャーププラントフェア'90 OSAKAスタジアム
に多数のご来場頂きまして、誠にありがとうございました。
アイ・ツ-サックスフェアPart2も只今企画中です。この
ご期待!!

X68000ユーザーとおきのグッズ!!
X68000ユーザーのステータスシンボル。
新グッズもグループインしてますます充実。
キミのパソコンライフが一層楽しくなるコレクションだ!
X68000オリジナルグッズをまだ持っていないキミ
アイ・ツ-からお届けしちゃいマス!



■営業時間 AM11:00~PM8:00

年中無休

Information & Interface
株式会社 アイ・ツ-

大阪店/〒542 大阪市中央区難波千日前15-18

パソコン専門

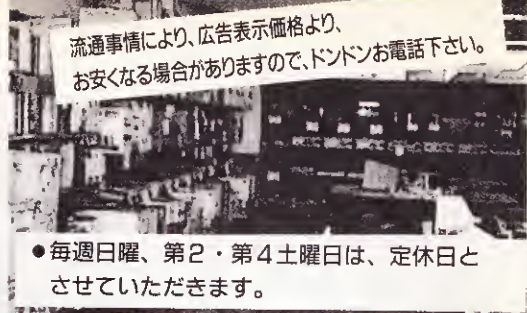
アフターサービス万全
のサポート体制
優良パソコン販売店

O.A.ランド

- お近くの方は、お立ち寄り下さい。
専門係員がアドバイスいたします。
- ビジネスソフト、ゲームソフトのこと
ならおまかせ下さい!!

セール期間
◀ '90 7・15 ▶ 8・15

サマーセール!! ドカ〜ンとプレゼント
OAランド恒例・大お買得セール実施中



●毎週日曜、第2・第4土曜日は、定休日と
させていただきます。

SHARP X68000シリーズセット (お楽しみゲームバック付)

●次代のインテリジェンス= SX-WINDOW搭載!!

X68000 EXPERT II

- CZ-603C-BK/GY
- CZ-605D-BK/GY
- MD-2HD 20枚

定価合計 ¥453,000

X68000 EXPERT II-HD

- CZ-613C-BK/GY
- CZ-605D-BK/GY
- MD-2HD 20枚

定価合計 ¥563,000

OAランド大特価

クレジット 12回 ¥30,200 24回 ¥15,900

NEW

OAランド大特価

クレジット 12回 ¥37,400 24回 ¥19,700

●SX-WINDOW搭載!!

X68000 PRO II

- CZ-653C-BK/GY
- CZ-605D-BK/GY
- MD-2HD 20枚

定価合計 ¥400,000

X68000 PRO II-HD

- CZ-663C-BK/GY
- CZ-605D-BK/GY
- MD-2HD 20枚

定価合計 ¥510,000

OAランド大特価

クレジット 12回 ¥26,600 24回 ¥14,000

OAランド大特価

クレジット 12回 ¥33,800 24回 ¥17,800

●SX-WINDOW搭載!!

X68000 SUPER-HD

- SX-WINDOW搭載
- SCSIインターフェース装備
- 80MBハードディスク搭載
- 3MB大容量メモリ装備
- 高解像度グラフィック

X68000 SUPER-HD

- CZ-623C-TN(チタン)
- CZ-613D-TN(チタン)
- MD-2HD 20枚

定価合計 ¥633,000

クレジット 12回 ¥40,600 24回 ¥21,400

NEW

OAランド大特価

X-1ターボZⅢセット

①セット

- CZ-88FCBK...定価¥169,800
- CZ-880DBK...定価¥109,800
- CZ-6ST1B...定価¥5,800 (チャイルドスタンド)
- MD-2HD 20枚サービス

合計定価¥275,400

特価中TEL下さい

②セット

- CZ-888CBK...定価¥169,800
- CZ-830DBK...定価¥98,000
- CZ-6ST1B...定価¥5,800 (チャイルドスタンド)
- MD-2HD 20枚サービス

合計価格¥273,600

特価中TEL下さい



今月の特価品(限定)お早目に!!

★CZ-652C(BK)+CZ-602D(BK)

4セット限り...大特価¥258,000

★CZ-612C(BK)

3セット限り...大特価¥298,000

- SHARP WD-A300(ワープロ) 定価¥165,000...特価¥110,000
- SHARP WD-A330(ワープロ) 定価¥185,000...特価¥125,000
- SHARP WD-HL30(ワープロ) 定価¥198,000...特価¥134,000
- SHARP PW-910(ワープロ) 定価¥85,000...特価¥85,000

- NEC PC-KD853(アナログCRT) 定価¥50,000...特価¥50,000
- 三菱XC-1498C(アナログCRT) 定価¥54,800...特価¥54,800
- SHARP QJ-48FD(アナログCRT) 定価¥46,000...特価¥46,000
- SHARP PA-8500(電子手帳) 定価¥16,000...特価¥16,000

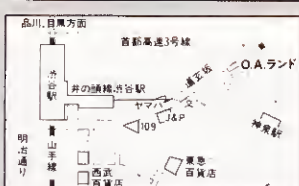
通信販売のご案内

全国通販

●銀行振込で申し込みの方は商品名
及びお客様の住所・氏名・電話番号
をお知らせ下さい。

[振込先]第一勧業銀行 渋谷支店
普通No.1163457 株オーエーランド

●現金書留で送金されるお客様は電話番号と商品名、数量を明記して同封して下さい。
●クレジットでご購入を希望される方は申し込み用紙をお送り致しますのでご記入の上返送して下さい。20日以上の方は、原則として保証人不要です。クレジットは1〜60回払で月々5,000円より自由に設定できます。



周辺機器コーナー

プリンターセットコーナー

- CZ-6PV1(カラービデオプリンター) 定価¥198,000...特価¥152,000
- CZ-8PC3(24ピン熱転写カラープリンター) 定価¥65,800...特価¥53,000
- CZ-8PK10(24ピン漢字ドットプリンター・135dpi) 定価¥97,800...特価/TEL下さい!
- CZ-8PG1(24ピンカラー漢字ドットプリンター・80dpi) 定価¥130,000...特価/TEL下さい!
- CZ-8PG2(24ピンカラー漢字ドットプリンター・135dpi) 定価¥160,000...特価/TEL下さい!
- IO-735X(カラーイメージジェットプリンター) 定価¥248,000...特価/TEL下さい!

OAランド特選品!!



■CZ-8PC4(定価 ¥99,800)

●48ピン熱転写カラー漢字プリンター 特価¥64,800

X68000用ソフトウェア・コーナー

- ① CZ-212BS(BUSINESS).....定価¥58,000▶特価¥53,000
- ② CZ-220BS(DATA).....定価¥58,000▶特価¥45,000
- ③ CZ-215MS(Sampling).....定価¥17,800▶特価¥13,800
- ④ CZ-221HS(NEW Print Shop).....定価¥10,800▶特価¥15,500
- ⑤ CZ-227BS(TOP財務会計).....定価¥200,000▶特価¥158,000
- ⑥ CZ-226BS(CARD).....定価¥229,800▶特価¥23,000
- ⑦ CZ-223CS(Communication).....定価¥19,800▶特価¥115,500
- ⑧ CZ-213MS(MUSIC).....定価¥18,800▶特価¥14,800
- ⑨ CZ-211LS(C compiler).....定価¥39,800▶特価¥31,000
- ⑩ C-TRACE(キャスト).....定価¥68,000▶特価¥52,000
- ⑪ EW(イースト).....定価¥38,000▶特価¥29,000

X68000用周辺機器コーナー

- CZ-6PU1A...定価¥38,000▶特価¥30,000
- CZ-6BM1...定価¥26,800▶特価¥21,000
- CZ-6BE1...定価¥68,000▶特価¥69,800
- CZ-6VT1...定価¥69,800▶TEL下さい!
- CZ-6SN1...定価¥188,000▶特価¥149,000
- CZ-6BC1...定価¥79,800▶特価¥63,000

●最新ゲームソフト
その他各種ソフト
20%〜25%OFF!!
●周辺機器・プリンター
割引販売中!! TEL下さい!

I-O DATA 増設RAMボード

●1MB増設RAMボード
PIO-6BE1-A

定価 ¥25,000

●2MB増設RAMボード
PIO-6BE2-2M

定価 ¥50,000

●4MB増設RAMボード
PIO-6BE4-4M

定価 ¥88,000

特価¥19,500 特価¥38,500 特価¥67,000

ハードディスク ■特価品もありますので TEL下さい。

- アイテック ITX-640.....特価¥117,000
- アイテック ITX-680.....特価¥149,000
- ロジテック LHD-32V.....特価¥85,000
- ロジテック LHD-34VE.....特価¥90,000
- ロジテック LHD-34V.....特価¥104,000
- シャープ CZ-620H.....特価¥118,000
- シャープ CZ-64H.....特価¥95,000
- アイテム HXD-040.....特価¥88,000
- アイテム HXD-042.....特価¥95,000
- ICM SR-80.....特価¥130,000

中古パソコン (価格/在庫は変動します。予約は5日以内とします。)

- | | |
|---------------------------|-----------------------------|
| PC-9801RA5.....¥338,000 | PC-286VS.....¥165,000 |
| PC-9801RA2.....¥265,000 | CZ-600C.....¥160,000 |
| PC-9801RX2.....¥199,000 | CZ-601C.....¥170,000 |
| PC-9801EX2.....¥190,000 | CZ-611C.....¥198,000 |
| PC-9801VX21.....¥170,000 | CZ-652C.....¥178,000 |
| PC-9801LUX21.....¥165,000 | CZ-612C.....¥210,000 |
| PC-9801VX2.....¥160,000 | 68000用モニター.....¥49,000 |
| PC-9801VM21.....¥150,000 | PC-9801用サウンドボード.....¥13,000 |
| PC-9801UV11.....¥148,000 | PC-88SR,FR.....¥50,000 |
| PC-9801LV22.....¥160,000 | PC-88FH,FA.....¥65,000 |
| PC-286VE.....¥150,000 | 400ラインCRT.....¥38,000 |
| PC-286US.....¥155,000 | 200ラインCRT.....¥10,000 |

- 取・買取は電話で見積りしております。責任を持って下取りさせていただきます。
- ご注文、お問合せは... 午前10時から午後7時まで
- 商品のお届けは...入金確認後、即日発送致します。

株オーエーランド

〒150 東京都渋谷区円山町20-4 第5日新ビル1F

☎(03)770-8855 FAX (03)770-7080

関東エリアの送料は、1個につき¥1,000です。
★全商品保証書付。専門のアドバイザーが、お客様のニーズに対応します。
★初期不良・輸送トラブル等に迅速に対応し、即交換させていただきます。

■表示価格は、税別表示です。詳しくは、お電話にて、お問い合わせ下さい。掲載の価格は、6月末現在です。

株式
会社

デンキヤ



営業時間AM11:00~PM7:00 水・木曜定休

セット超特価

68000

PERSONAL WORKSTATION

PRO II・PRO II HD

CZ-653C特価

CZ-663C特価

SUPER HD

CZ-623C特価

CZ-613D特価

(価格はすべて税込みです)

セット超特価

68000

PERSONAL WORKSTATION

EXPERT II・EXPERT II HD

CZ-603C特価

CZ-613C特価

EXPERT PRO

CZ-662C特価

CZ-602C特価

全品メーカー保証 即決クレジットOK

ディスプレイ

CZ-604D 特価
CZ-605D 特価
CZ-613D 特価
CU-21HD 特価

プリンタ

CZ-8PC4 特価
CZ-8PG1 特価
CZ-8PG2 特価
IO-735X 特価

周辺機器

CZ-8NJ1 ¥1,400
CZ-8NJ2 ¥18,540
PIO-6BE1A ¥20,000
PIO-6BE2 ¥39,000

ソフト

CZ-213MS ¥15,500
CZ-223CS ¥15,300
CZ-219SS ¥23,100
CZ-211LS ¥30,800

24時間テレホンサービス

0482-54-3444

お申し込み

TEL.0482-54-3400

FAX.0482-54-3443

埼玉県川口市西川口4-6-4

お支払い

下記取引銀行口座

までお振込み下さい。

三菱銀行西川口支店

(株)デンキヤ ⑨0258081



クリエイイト特典

- 全商品完全保証書付(メーカー保証)
- 全国無料配達(一部離島の方は有料になります)
- 配達日の指定OK(日曜・祭日にかかわらずお客様のご都合にあわせて配達します)
- どんな商品の組合せも自由自在(ご予算、用途に応じ自由自在にシステムアップできます)
- 中古パソコン高額下取り(今お使いのパソコンをわずかな差額でグレードアップ)
- お支払い方法自由(低金利の均等払い、ボーナス一括払いもご利用ください)

営業時間(定休日▶渋谷店:日曜・祭日/横浜店:水曜)
AM10:00~PM7:00

当社はX68000の販売認定店です。
どんなことでも安心してご相談ください。

X68000

ビッグ・サマーセール開催中!

即売・即納

X68000 NEW PRO II

- CZ-653C(本体).....¥285,000
- CZ-603D(カラーディスプレイ).....¥84,800
- 好きなゲームソフト1本.....¥7,800
- 定価合計.....¥377,600

クリエイイト特価

均等払い	¥7,680×48回	¥9,890×36回	¥14,370×24回
ボーナス	なし	なし	なし

X68000 NEW EXPERT II

- CZ-603C(本体).....¥338,000
- CZ-613D(カラーディスプレイテレビ).....¥135,000
- CZ-8NJ2.....¥23,800
- 好きなゲームソフト1本.....¥9,800
- 定価合計.....¥506,600

クリエイイト特価

均等払い	¥9,970×48回	¥12,840×36回	¥18,660×24回
ボーナス	なし	なし	なし

X68000 EXPERT II HD

- CZ-613C(本体).....¥448,000
- CZ-604D(カラーディスプレイ).....¥94,800
- 好きなゲームソフト1本.....¥9,800
- 定価合計.....¥552,600

クリエイイト特価

均等払い	¥5,920×48回	¥7,400×36回	¥12,100×24回
ボーナス	¥30,000×8回	¥40,000×6回	¥50,000×4回

X68000 SUPER HD

- CZ-623C-TN(本体・キーボード・マウス).....¥498,000
- CZ-613D-TN(カラーディスプレイ).....¥135,000
- CZ-6BP1.....¥79,800
- 定価合計.....¥712,800

クリエイイト特価

均等払い	¥7,320×48回	¥10,100×36回	¥13,450×24回
ボーナス	¥42,000×8回	¥50,000×6回	¥80,000×4回

※本広告に掲載の全商品の価格について消費税は含まれておりません。

X68000 NEW EXPERT II

ミュージシャンセット。これもTMネットワークだよ〜!

- CZ-603C.....¥338,000
- CZ-605D.....¥115,000
- MU1.B(MIDIボード&ソフト).....¥39,800
- CM32L.....¥69,000
- グラナダ.....¥8,800
- JOYカード.....¥1,800
- 定価合計.....¥572,400▶超特価¥458,000

X68000 NEW PRO II

ゲームズセット。遊んで暮らせるSET!

- PRO II CZ653C.....¥285,000
- 0.31CRT CZ603D.....¥84,800
- グラナダ.....¥8,800
- Y'S.....¥8,700
- ポピュラス.....¥9,800
- スーパーハンガオン.....¥8,800
- エージャックス.....¥8,800
- サーク.....¥8,800
- アールタイプ.....¥7,800
- アナログJOYSTIC XE-IAP.....¥13,800
- 定価合計.....¥445,100▶超特価¥353,000

X68000シリーズ用 周辺機器・ソフト オール超特価!!

型番	品名	定価	ソフト名	品名	定価
CZ-6VT1	カラーイメージユニット	¥69,800	MUSIC PRO	MIDI版	¥28,800
CZ-8NS1	カラーイメージスキャナ	¥188,000	MUSIC PRO-68K	マウスを使った楽譜ワープロ	¥18,800
CZ-6BE1A	IMB増設RAMボード	¥38,000	SOUND PRO-68K	サウンドエディタ	¥15,800
CZ-6BE2	2MB増設RAMボード	¥79,800	Sampling PRO-68K	AD PCMサンプリングエディタ	¥17,800
CZ-6BE4	4MB増設RAMボード	¥138,000	Musicstudio PRO-68K V1.1	MIDIマルチレコーディングソフト	¥28,800
CZ-8NM3	マウス・トラックボール	¥9,800	OS-9 X68000	マルチタスクオペレーティングシステム	¥23,800
BF-68PRO	高性能CRTフィルター	¥19,800	PRO-68K	サイバーノート	¥13,800
CZ-6BP1	数値演算プロセッサボード	¥79,800	PRO-68K	ステーションナリー	¥14,800
CZ-8NT1	トラックボール	¥13,800	Ccompiler PRO-68K	ソフト開発セット	¥39,800
CZ-6BM1	MIDIボード	¥26,800	Human 68K Ver2.0	開発ツールセット	¥9,800
CZ-8NJ2	アナログスティック	¥23,800	PIO-6BE1-A	内蔵1MRAM	¥25,000
CZ-6TU	パソコンチューナ	¥33,100	PIO-6BE2-2M	2MRAM	¥50,000
SX-68M	MIDI I/F	¥19,800	PIO-6BE4-4M	4MRAM	¥88,000
XE-IAP	アナログジョイパッド	¥13,800	MU1-B	MIDI I/F+ソフト	¥39,800

▲上記以外ビジネスソフト、最新ゲームソフト豊富に在庫あります。※送料はご注文の際お問合せください。●超特価販売中!

オール15%~20%OFF

総合お問合せ先☎03-486-6541代

パソコン専門ショップ

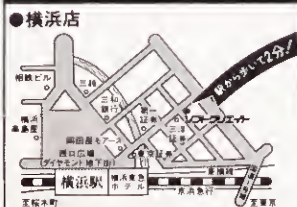
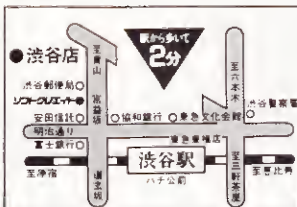
ソフトクリエイイト 渋谷/横浜

●渋谷店☎03-486-6541(代)

〒150 東京都渋谷区渋谷1-12-7 三和渋谷ビル
振込銀行:三井銀行 渋谷宮益坂支店①No.5000340

●横浜店☎045-314-4777(代)

〒221 横浜市神奈川区鶴屋町2-12-8 第1建設ビル
振込銀行:三和銀行 横浜駅前支店①No.310852



X68000 全機種取り揃え大特価セール

68000 EXPERT/PRO



CZ-602C (本体)
+ CZ-603D (ディスプレイ)
+ SX.WINDOW
大特価 ¥310,000
(このセッに限り、送料+消費税込)
CZ-653C (本体)
+ CZ-602D (ディスプレイ)
大特価 ¥288,000
(このセッに限り、送料+消費税込)

※代金は商品引換着払いでもOKです。

New X68000新発売 / (●特価価格は直接お問合せください)

CZ-603C	定価 ¥338,000	<ディスプレイ>	
CZ-613C	定価 ¥448,000		定価 ¥84,800
CZ-623C	定価 ¥498,000	CZ-604D	定価 ¥94,800
CZ-653C	定価 ¥285,000	CZ-605D	定価 ¥115,000
CZ-663C	定価 ¥395,000	CZ-613D	定価 ¥135,000

●新製品も
大特価 / お
問合せくだ
さい。

SHARPラップトップパソコン AX286L-F



お買得品
定価 ¥428,000 →
特価 ¥238,000

ALBIT

アイビット電子株式会社

FM TOWNS お買い得セット



大特価 / ¥285,000

MZ2500下取り / MZ2500からMZ2861 (定価 ¥328,000) へ買い替え下取後 特価 ¥165,000
CZ600C下取り / CZ600CからCZ623 (X68000 SUPER) へ買い替え下取後 特価 ¥300,000

ハガキもOK、New MZプリンタ シャープMZ-IP22

熱転写プリンタ
●好評発売中 /
●24 x 24dot/英寸 ●7色カラー ●
●30dot/英寸 ●高速印刷 ●MZ1P
17dot/英寸 ●50dot/英寸 ●50dot/英寸
●メモリ付 ●対応OS: MS-DOS, MZ2000,
2500, 5500, 6500シリーズ, X1シリ
ーズ, X68000シリーズ他

標準価格 ¥59,800 → **特価 ¥25,000**

パソコンファクスMZ-1V01

●"プリンタ・コピー・ファクス"
1台3役のスクレモノ
●MZ-1V01 (本体のみ)
標準価格合計 ¥342,800 → **¥120,000**
●MZ-1V01 (本体のみ)
標準価格合計 ¥278,000 → **¥98,000**

シャープMZ-1X30 モデムホン (1×19上位機種)

●300/1200bps全2重通信対応
モデム内蔵 ●音声入出力端子
●ダイヤルパルス / フッシュボ
タン対応 ●フッシュボタン音解
機能 ●シャープ手帳、COIT、V25
bis通信手帳サポート

標準価格 ¥98,000 → **大特価**

パソコンと専用ワープロをひとつにした18ビット シャープMZ-2861

ワープロソフト「書院28」
MS-DOS V3.1 装備
エミュレーションソフト搭載
定価 ¥328,000 →
大特価 ¥198,000
(ディスプレイは別)

MZ-2861用ソフト(UPシリーズ)

●IP-1251 (デスクトップ) 定価 ¥88,000 → 特価 ¥20,000
●IP-1253 (キーボード) 定価 ¥77,000 → 特価 ¥20,000
●IP-1254 (プリンタ) 定価 ¥89,000 → 特価 ¥20,000

シャープMZ-2520

定価 ¥159,800 → **大特価 ¥78,000**

《在庫限り》 PC-E500PJ

定価 ¥28,800 → **大特価**
SHARPポケコン全機種、
Z80ボード他、太平洋工業製品全機種取扱
●PC-E500PJに購入の方に、もちろん
「ポケコンシャープナル特別号」を進呈 /
PC-500と各種パソコンをつなぐインターフェースケーブル
CE-140T ¥8,800

拡張機器他

- シャープCZ-8GR (286拡張) ¥32,000 → ¥12,000
- シャープCZ-8B3 (386拡張) ¥33,800 → ¥28,000
- シャープCZ-8B3 (X1) ¥13,800 → ¥11,700
- シャープCZ-8B4 (X1) ¥6,800 → ¥5,700
- シャープCZ-8BGR2 (X1) ¥14,800 → ¥4,000
- シャープCZ-64H (286拡張) 特価
- シャープCZ-8N2 (286拡張) ¥23,800 → ¥18,500
- シャープCZ-8E1 ¥38,000 → ¥25,000
- シャープCZ-8T (286拡張) ¥8,500 → ¥1,000
- シャープMZ-1U08 ¥25,000 → ¥12,000
- シャープMZ-1U03 ¥35,000 → ¥15,000
- シャープMZ-1X22 (286拡張) ¥21,800 → ¥13,000
- シャープMZ-1R12 (286拡張) ¥35,000 → ¥8,000
- シャープMZ-1E29 (286拡張) ¥17,800 → ¥9,800
- シャープMZ-1E30 (286拡張) ¥25,000 → ¥22,500
- シャープMZ-1U09 (2500) ¥9,000 → ¥7,200
- シャープMZ-1M03 (5500) ¥69,000 → ¥35,000
- シャープMZ-8B04 (2000) ¥18,000 → ¥8,000
- シャープMZ-8B104 (2000) ¥45,000 → ¥18,000
- シャープMZ-1R11 (5500) ¥80,000 → ¥30,000
- シャープMZ-1R24 (1500) ¥22,000 → ¥6,000
- シャープMZ-1R26A (2500) ¥13,000 → ¥12,800
- シャープMZ-1R27A (2500) ¥13,000 → ¥10,000
- シャープMZ-1R28A (2500) ¥13,000 → ¥10,000
- シャープMZ-1R29A (2500) ¥32,000 → ¥10,000
- シャープMZ-1T02 (2200) ¥19,800 → ¥8,500
- シャープMZ-1T03 (1500) ¥12,000 → ¥8,500
- シャープMZ-1X29 ¥13,800 → ¥11,000
- フレックスRM-2SE (286拡張) ¥428,800 → ¥38,500
- シャープCZ-65T1 (286拡張) 特価 ¥3,500
- シャープCZ-8B2 (286拡張) ¥29,800 → ¥25,300
- シャープCZ-8B2 (286拡張) ¥19,800 → ¥16,800
- シャープX1 MZ用マウス 特価 ¥4,800
- シャープX1用ジョystick ¥1,500
- シャープMZ-5500キーボード ¥8,000
- シャープ2000/2200キーボード ¥8,000
- シャープMZ-1E08 ¥9,000 → ¥8,000
- シャープCZ-6B1 (286拡張) ¥26,800 → ¥23,000
- シャープCZ-6B1 (286拡張) ¥35,000 → ¥29,500
- シャープCZ-6B1A (286拡張) ¥38,000 → ¥23,800
- 7インチFD (286拡張) ¥25,000 → ¥21,500
- 7インチFD (286拡張) ¥50,000 → ¥42,500
- 7インチFD (286拡張) ¥88,000 → ¥74,500

(MZ-2861)

- シャープMZ1R35 (286拡張) ¥55,000 → ¥19,000
- シャープMZ1R36 (286拡張) ¥45,000 → ¥15,000
- シャープSS-SC28M (286拡張) ¥49,800 → ¥10,000
- シャープCZ-8PC1 (286拡張) ¥99,800 → 大特価
- シャープCZ-8PC2 ¥100,000 → ¥100,000
- シャープCZ-8PC2 ¥160,000 → ¥130,000
- シャープMZ-1P27 ¥268,000 → ¥214,400
- シャープMZ-1P28 ¥148,000 → ¥118,400
- シャープMZ-1P29 ¥168,000 → ¥134,400
- シャープMZ-6P18 ¥60,000 → ¥35,000
- シャープMZ-6P27 ¥58,000 → ¥39,800
- シャープMZ-6P27 ¥50,000 → ¥37,500

フロッピーディスク

- シャープCZ-501H (286拡張) ¥258,000 → ¥60,000
- シャープCZ-503F ¥49,800 → ¥30,000
- シャープCZ-53F ¥19,800 → ¥9,800
- シャープCZ-300F (CZ-3PC付) ¥13,000

ハードディスク

- アイテックIT-X640 ¥158,000 → ¥128,000
- アイテックIT-X68 ¥198,000 → ¥158,000

ディスプレイ

- シャープMZ-1D17 (286拡張) ¥124,000 → ¥63,000
- シャープMZ-1D27 ¥120,000 → ¥79,800

ソフト

- X68000用
- CZ-230AS ニューゼaland ¥8,800 → ¥7,040
- CZ-230AS FULL THRILL ¥8,800 → ¥7,040
- CZ-233AS PACMANIA ¥7,800 → ¥6,250
- CZ-222AS ARKANOID ¥7,800 → ¥6,250
- POPULOUS ¥9,800 → ¥7,850
- CZ-239AS THUNDERBLADE ¥9,500 → ¥8,000
- CZ-259SS X68000XWindow 特価

(MZ-2500用)

- IP-1215 COBOL ¥13,800 → ¥11,700
- DANGER BOX ¥5,800 → ¥2,000
- EXTRA HYPER DISK MONITOR ¥10,000 → ¥8,500
- EXTRA HYPER DISK MONITOR ¥14,000 → ¥12,000
- FILE UTILITY CUT-25P ¥6,800 → ¥6,000
- FREE CALL ¥6,800 → ¥1,000
- G-ED12500 ¥8,000 → ¥7,000
- H.S.コンローラ ¥9,600 → ¥8,500
- HUCAL日本語 ¥45,000 → ¥15,000
- カレイドスコープ ¥9,800 → ¥3,000
- カレイドスコープ2 ¥9,800 → ¥1,000
- ザ・ブラッックオニキス ¥7,800 → ¥3,000
- スーパー修理屋さん ¥12,000 → ¥10,200
- ムーンチャイルド ¥7,800 → ¥3,000
- 英雄伝説サガ ¥9,800 → ¥2,000
- 五弁並べ ¥4,800 → ¥2,000
- 探検隊最強 ¥7,800 → ¥2,000
- プリントSHOP ¥9,800 → ¥8,500
- プリントSHOPライブラリー ¥4,500 → ¥3,800
- プリントSHOPライブラリー2 ¥4,500 → ¥3,800

(X1用)

- 日本語ワープロ将軍X1 ¥34,800 → ¥29,000
- 日本語ワープロ侍X1 ¥19,800 → ¥16,800
- CZ-8W51 X12システム ¥9,800 → ¥3,500
- 3CP/M X1 3CP/M ¥16,800 → ¥11,000
- CZ-8B3 X11 第二水準ROM ¥13,800 → ¥11,700
- CZ-1285F X1 CP/M ¥13,800 → ¥11,700
- CZ-1305F X1 CP/M ¥14,800 → ¥12,500
- CZ-1161F X1 C ¥13,800 → ¥11,700
- CZ-1175 X1 LOGO ¥18,800 → ¥13,200
- CZ-1181F X1 COBOL ¥13,800 → ¥11,700
- CZ-1261F X1 APL ¥13,800 → ¥11,700
- CZ-1151F X1 FORTRAN ¥11,700

(MZ-5500, 6500SOFT)

- MZ-22013 (MZ-5500MSDOS)
- MZ-22014 (MZ-5500TODAY)
- MZ-22023 (MZ-5500G.W.BASIC)
- MZ-22028 (MZ-5500G.W.BASIC)
- MZ-22029 (MZ-5500G.W.BASIC)
- MZ-22029 (MZ-5500TODAY)

本体 ●シャープCZ-820, 822, 880, 881, MZ-3500, 2520, 2861, 2200, X68000, CZ-612, 662, 602, 652 ●富士通FM-77AV-1, 77AV-2, 77AV-20, 77AV-40 ●NEC PC-9801N ●東芝J3100S

SHARPラップトップパソコン
All in Note 新発売! (入荷)
AX286N-H2 定価 ¥398,000

(全商品新品完全保証付) ■シャープポケコン全商品販売中。カタログ、特価表ご請求ください(〒72)

アイビット推奨ディスプレイ

●三菱XC-1498CII (14型アナログ) ドットピッチ0.28 定価 ¥107,000 → 特価 ¥59,800
XC-1498CII対応パソコン機種: PC-9801シリーズ / PC-286シリーズ / PC-386シリーズ / PC-8801シリーズ (上記機種には付属の接続ケーブルで、接続可能)

●シャープCZ-830D-BK (14型) 2モードオートスキャン方式 (アナログ/デジタル) 定価 ¥98,000 → 特価 ¥54,800
CZ-830D対応パソコン機種: CZ880C/881C, X1/TURBOシリーズ、ケーブルは本体付属を使用。NEC PC-8801/9801シリーズ(XA-XLのみ不可) MZ700/1500/2000/2200/2500各シリーズ (推奨品シャープ8D8K)。

●シャープCZ-602D-BK (15型アナログTV/3モードオートスキャン) 定価 ¥99,800 → 特価 ¥75,000
CZ-602D対応パソコン機種: ※X1シリーズ / ※X1 turboシリーズ / X1 yurboZシリーズ / X68000シリーズ / PC8801シリーズ / PC-9801シリーズ / PC-286シリーズ (※は接続ケーブルANI506が必要です)

●シャープCZ-603D-GY-BK (15型カラーディスプレイTV) ドットピッチ3.9 定価 ¥84,800 → 特価
CZ-603D対応パソコン機種: ※X1シリーズ / ※X1 turboシリーズ / X1 yurboZシリーズ / X68000シリーズ / PC8801シリーズ / PC-9801シリーズ / PC-286シリーズ (※は接続ケーブルANI506が必要です)

0426-45-3001~3
FAX.0426-44-6002

●営業時間/10:00~19:00 ●電話受付/20:00迄可 ●定休日/日曜日(祭日営業)

SHARP SUPER XEX SHOP

アイビット電子株式会社 〒192 東京都八王子市北野町560-5

●本誌発売時には上記価格よりさらにお求めやすい価格に変更されている場合があります。●一部を除き、上記商品価格には消費税は含まれておりません。その商品に対し別途3%の消費税がかかりますのでご了承ください。

上記の広告商品はすべて店頭販売もしております。

★送料はご注文の際にお問い合わせ下さい。

★掲載の商品は、すべて新品、保証書付きです。

★掲載の商品は充分用意してありますが、ご注文の際は、在庫の確認の上、現金書留または、銀行振込でお申し込み下さい。全商品クレジットでも扱っております。

★お申し込みの際は必ず電話番号を明記して下さい。

★商品、品切れの際はご容赦下さい。

北海道から沖縄まで

富士銀行八王子支店 (普) 1752505

PERSONAL COMPUTER MAGAZINE
PC-9801活用版

Oh!PC

パソコン
●
テスト
●
マガジン

PC
MAGAZINE
JAPANESE EDITION

THE
COMPUTER
コンピュータ時代を読む トレンド・マガジン

C MAGAZINE

ソフトバンク
の
雑誌が勢揃い

月刊・コンピュータ技術者必修
第2巻・第1巻・両巻受検

情報処理試験

Oh!FM

Oh!△

BEEP! POWERFUL MEGA-MAGAZINE
MEGADRIVE
メガドライブ

《広告の半ページ》 ああ、また暑い夏がくる。♪ヘナヘナヘーナーヘエナ〜 ちからがぬーけーるー

月刊 電脳倶楽部 90年8月号 (Vol.27) 7月18日発送

2HDディスクに入ったX68000のための雑誌だっ!

もしかして

CW送受信練習プログラム

さらに もしかして

X-BASIC配列サーチ・ソート関数

さらには

CZ-8PC2用
カラーハードコピープログラム

そして

しりとりPRO-68K

とどめはPDDで

軽犯罪法・日米安保条約

その他、便利なツール、PDD、ビーブ音、読み物などを満載!

(なお、内容は一部変更されることがあります。ご了承下さい)

編集長祝一平からの御挨拶「どもども、暑いですねえ。今日なんかアイスクャンデーを五本も食べてしまいました。五本といえば龍角散」

満開製作所 電脳倶楽部 編集部

〒171 東京都豊島区要町1-19-3 いさみビル4F
TEL.(03)554-9282/FAX.(03)554-3856

販売方法は通信販売のみです。お申し込みの方法は左記の住所へ現金書留で
定期購読 6ヶ月分 6,000円 (消費税込・郵送料サービス)

- 7月18日以降に受け付けた分は、原則としてVol.27から発送します。新たに購読を希望される方は、「新規」と御明記下さい。
- 郵便振替を御利用の場合は口座番号「東京5-362847 満開製作所」をお願いいたします。製品の性格上、返品には応じられませんが、お申し出があれば定期購読を解約し残金をお返します。(ご注意: バックナンバーの受け付けは、定期購読の方に限らせていただきます)

△V1 エミュレータ

好評発売中

定価¥9,800



X1エミュレータはX68000上でX1シリーズのアプリケーションを実行するためのソフトエミュレータです。X1のアプリケーションを完全にソフトウェアのみでエミュレートしているため、X1上での実行速度と比較して、平均3~5倍程度おそくなりますが、X68000のマシン上に実現した仮想X1マシンを楽しめます。また、X1とX68000の相互間でファイルを転送するためのユーティリティと専用ケーブルが付属しますので、X1上で作り上げたソフトの資産をX68000上に移行することも簡単にできます。

△V1 エミュレータの機能

- X1エミュレータはX1に相当する機能をエミュレート。
この仮想コンピュータには最大4つのドライブが仮想的に接続。
- X1エミュレータからみたドライブはHuman68kのドライブ上にあるファイルで仮想的に実現。このファイルはX1用の5" 2Dディスクのイメージをファイル転送ユーティリティでまるごと転送したもの。
- X1エミュレータで仮想的に実現したX1は仮想ドライブから起動。
このため仮想ドライブ用ファイルには、X1を立ち上げるために必要なHuBASICやCP/Mなどのシステムプログラムが必要。
- X1エミュレータでは、X1の持つVRAMを含むメモリーイメージやZ80CPUを仮想的にソフトウェアで実現。

ファイル転送ユーティリティ

ディスク転送

X1ディスク ↔ X68000 Human68k (5" 2Dディスクイメージファイル)

- X1エミュレータではHuman68k上のディスクイメージファイルを仮想ドライブとして使用。

ファイル転送

X1 BASIC: CP/M ↔ X68000 Human68k

- X1で作ったプログラム&データをX68000上で使用。
- ※ 付属の専用ケーブルをX1とX68000に接続してファイルを転送します。



△V1 エミュレータ Q&A

- Q. ファイル転送のために別途RS-232Cケーブルを買わないといけないのですか?
A. 専用のケーブルが付属しますのでその必要はありません。
- Q. X1BASICのプログラムをX68000上のX-BASICで使えますか?
A. 通常のセーブではコードが違うので使用できませんが、アスキーセーブしたファイルであればX-BASIC上でそのままロード可能です。
- Q. TurboBASICで作成した住所録などの漢字を含んだデータがあるのですがX68000上にファイル転送できますか?
A. X1TurboもX68000も漢字はシフトJISコードなのでファイルの転送は可能です。ただし、漢字ROMを必要とするものはサポートしていません。
- Q. Turbo用のソフトは動きますか?
A. X1用のみでTurbo専用のソフトは動きません。
- Q. ゲームは動きますか?
A. 純粋にBASICでかかれたものは動きますが、プロテクトがかかったものや直接ハードをアクセスするような市販のゲームは動きません。
- * タイミング等ハードウェアに依存するようなソフトは、原理上実行できない、もしくは正常に動作しない場合がありますのでご注意ください。
* 一部サポートしていない機能があります。
- X1エミュレータ通信販売** 購入希望として住所、氏名、電話番号をお知らせください。注文書をお送り致します。

* この商品価格には消費税は含まれておりません。

* CP/Mはデジタルリサーチ社の商標です。

文中のソフトウェアは各社の商標です。

* 製品の仕様、名称は予告なく変更する場合がございますのであらかじめご了承ください。

有限会社 **アクセス** 〒101 東京都千代田区神田神保町1-64
03 (233) 0200 (代) FAX. 03 (291) 7019
神保町協和ビル7F



たいまそくを
通信メディアと仕事の
開発をつくりだしています

-その他楽しいメニューがまだまだいっぱい！-

- ★ J & P ならではのパソコン・家電製品の会員割引もある**ONLINE SHOPPING**。
- ★ J & P だから強い**パソコン情報**をはじめとする役に立つ**DATA BASE**。
- ★ みんなでおしゃべり**オンライントーク**（CHAT 機能）。
- ★ 地域別・テーマ別ボードで充実の**BBS**（電子掲示板）。
- ★ ビジュアルデータもばっちり送受信できる**X-MODEM**。

J&P HOT LINEへのご入会はスタータキットで。

買ったその日から
2週間無料で
アクセスできます。

[illegible]

お問い合わせは JSP HOT LINE まで

スタータキットのお求めはJ&P各店でどうぞ。

洪各店	東京都渋谷区道玄坂2丁目28番4号	(03) 496-4141
町田店	東京都町田市青井1丁目39番16号	(0427) 23-13
八王子店	東京都八王子市旭町1番1号4-3分	(0426) 26-4141
立川店	東京都立川市幸町4-39-1	(0425) 36 4141
厚木店	厚木市中町3-4-3	(0462) 125-1548
富士店	富士市桜町2-1-10	(0764) 32-3133
金沢店	金沢市入江2-63	(0762) 91-1130
寺田店	金沢市寺田2-3	(0762) 147-2524
大須店	名古屋市中区大須2丁目2-48	(052) 262-1141

テクノラント	大阪市 東淀区 日ノ塚5丁目 番1番	06-654-1111
メリアント	大阪市 東淀区 日ノ塚5丁目 番1番	06-654-1111
コスモラント	大阪市 東淀区 日ノ塚5丁目 番1番	06-654-1111
U.S.LAND	大阪市 東淀区 日ノ塚4丁目 番1番	06-654-1111
ビネシラント	大阪市 東淀区 日ノ塚4丁目 番1番	06-654-1111
梅田 店	大阪市 北区 梅田 1丁目 番1番	06-561-1111
くすは 店	大阪市 東淀区 日ノ塚5丁目 番1番	06-654-1111
千里中央 店	大阪市 東淀区 日ノ塚5丁目 番1番	06-654-1111
川津富田 店	大阪市 東淀区 日ノ塚5丁目 番1番	06-654-1111
曙星 店	大阪市 東淀区 日ノ塚5丁目 番1番	06-654-1111

[illegible]

ADVANCED TURBO

先駆の“Z”アビリティがパソコンクリエイターを魅了する。



AV1 パソコンテレビ turbo Z III

パーソナルコンピュータ+キーボード+マウス	CZ-888C-BK	標準価格	169,800円(税別)
14型カラーディスプレイテレビ	CZ-860D-BK	標準価格	92,200円(税別)
チルトスタンド	CZ-6ST1-B	標準価格	5,800円(税別)

クリエイティブマインドを刺激するAV機能 テレビ、ビデオ、ビデオディスクなどの映像を最大4,096色のリアルな画像で瞬時にグラフィック画面に取り込めるカラー画像デジタイズ機能を標準装備。4段階の量子化取り込み、42通りのモザイク取り込みなど多彩なトリック取り込み処理もサポート。さらにクロマキー合成、インターレーススーパーインポーズ、4,096色対応デジタルテロップ機能、ステレオFM音源…先駆のAV機能がアートワークの領域をさらに拡げます。

AV指向の高水準ベーシックZ-BASIC搭載 多色グラフィック、カラー画像処理、ステレオFM音源、バンクメモリ対応など、ターボZシリーズが本来もつクリエイティブな機能をフルサポート。また豊富な画面モードで多色を駆使するときに便利なグラフィック用関数(HSV、RGB、HALF、CDOWN、CUP)も装備。さらにFM音源制御用ステートメントとしてX68000と命令コンパチの拡張MMLの採用によりスムーズな8音同時演奏を実現しています。

●メインメモリ128Kバイト標準装備、Z-BASICで最大576Kバイトまでサポート ●1Mバイトの5インチフロッピーディスクドライブ2基搭載 ●JIS第1/第2水準標準漢字、「システム・ユーザー辞書」を標準装備した高度な日本語処理機能 ●ニューデザインのマウス標準装備 ●X1ターボシリーズの豊富なソフト資産が活用できるコンパチブル設計 ●プリンタ、RS-232Cなど豊富なインターフェイスを装備 ●ドットピッチ0.39mmのハイコントラストブラウン管、15kHz/24kHzのデュアルスキャン方式採用14型カラーディスプレイテレビ(別売)。